Implementation of a Localization System for Sensor Networks

Tufan Coskun Karalar
Jan M. Rabaey

Electrical Engineering and Computer Sciences
University of California at Berkeley

May 18, 2006

Report Documentation Page		*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **18 MAY 2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**
4. TITLE AND SUBTITLE **Implementation of a Localization System for Sensor Networks**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Berkeley,Electrical Engineering and Computer Sciences,Berkeley,CA,94720**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Localization is very important for self-configuring wireless sensor networks. There are two main tasks to performing localization. Assuming availability of reference points, first the relationships to the reference points are established; in this thesis this relationship is the distance to the reference point. Second, using the reference point positions and the relations to these points, an algorithmic computation is carried out to compute the position. In the existing body of research on sensor network localization, the algorithmic aspects of this final position calculation have received the most attention. However there remain significant implementation issues related to both distance measurements and algorithmic computations. In this thesis the implementation issues regarding a sensor network localization system is studied along with some examples. In the first half, the implementation of a distributed, least-squares-based localization algorithm is presented. Low power and energy dissipation are key requirements for sensor networks. An ultra-low-power and dedicated hardware implementation of the localization system is presented. The cost of fixed-point implementation is also investigated. The design is implemented in a 0.13? CMOS process. It dissipates 1.7mW of active power and 0.122nJ/op of active energy with a silicon area of 0.55mm2. The mean calculated location error due to fixed-point implementation is shown to be 6%. In the second part, a radio frequency(RF) signal based Time of Flight (ToF) measuring ranging system for wireless sensor networks is proposed, designed and prototyped. The prototype measurement error is within -0.5m to 2m while operating at 100Msps sampling rate and using a 50MHz signal in the 2.4GHz ISM band. The system accuracy is limited by the sampling rate and can be linearly improved with increasing rates. This RF method is more cost e?ective than acoustic signal based ranging schemes, as it does not require ultrasonic transducers. The system is multipath resilient and can coexist with 2.4GHz band devices such as 802. 11b/g networks. The estimated power consumption for the digital baseband is 2.35mW and its estimated area 0.25mm2 in a 90nm CMOS process.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**173**	

Implementation of a Localization System for Sensor Networks

by

Tufan Coskun Karalar

B.S. (University of Michigan, Ann Arbor) 2000
M.S. (University of California, Berkeley) 2002

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:
Professor Jan M. Rabaey, Chair
Professor Kristofer S. J. Pister
Professor Paul K. Wright

Spring 2006

The dissertation of Tufan Coskun Karalar is approved:

_____

Chair   Date

_____

Date

_____

Date

University of California, Berkeley

Spring 2006

Implementation of a Localization System for Sensor Networks

Copyright 2006

by

Tufan Coskun Karalar

Abstract

Implementation of a Localization System for Sensor Networks

by

Tufan Coskun Karalar

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Jan M. Rabaey, Chair

Localization is very important for self-configuring wireless sensor networks. There are two main tasks to performing localization. Assuming availability of reference points, first the relationships to the reference points are established; in this thesis this relationship is the distance to the reference point. Second, using the reference point positions and the relations to these points, an algorithmic computation is carried out to compute the position. In the existing body of research on sensor network localization, the algorithmic aspects of this final position calculation have received the most attention. However there remain significant implementation issues related to both distance measurements and algorithmic computations.

In this thesis the implementation issues regarding a sensor network localization system is studied along with some examples. In the first half, the implementation of a distributed, least-squares-based localization algorithm is presented. Low power and energy dissipation are key requirements for sensor networks. An ultra-low-power and dedicated hardware implementation of the localization system is presented. The cost of fixed-point implementation is also investigated. The design is implemented in a 0.13μ CMOS process. It dissipates 1.7mW of active power and 0.122nJ/op of active

energy with a silicon area of 0.55mm^2. The mean calculated location error due to fixed-point implementation is shown to be 6%.

In the second part, a radio frequency(RF) signal based Time of Flight (ToF) measuring ranging system for wireless sensor networks is proposed, designed and prototyped. The prototype measurement error is within -0.5m to 2m while operating at 100Msps sampling rate and using a 50MHz signal in the 2.4GHz ISM band. The system accuracy is limited by the sampling rate and can be linearly improved with increasing rates. This RF method is more cost effective than acoustic signal based ranging schemes, as it does not require ultrasonic transducers. The system is multipath resilient and can coexist with 2.4GHz band devices such as 802.11b/g networks. The estimated power consumption for the digital baseband is 2.35mW and its estimated area 0.25mm^2 in a 90nm CMOS process.

Professor Jan M. Rabaey
Dissertation Committee Chair

To my parents Yalçın and Şafak Karalar

and

my brother Ömer Aytuğ Karalar,

Contents

List of Figures

List of Tables

Acknowledgments

The past six years I spent in Berkeley are no doubt going to be the most memorable years of my life. The main reason for this is the people I have gotten to know and befriend during my studies here. Without them Berkeley would not have been the same place..

First acknowledgement is for my advisor Jan Rabaey. His vision and insight was instrumental to my research and its success. His trust and confidence on me fills me with pride. Jan also needs credit for his part in teaching me the basics of circuit design, setting up the best research environment a grad student could ask for and being a role model for me.

Professor Kris Pister is acknowledged for his many contributions to my dissertation as a very interested thesis committee member. Professor Paul Wright, in addition to being a skilled instructor in his Rapid prototyping class, was a very helpful member of my committee. I would also like to thank Professor Bob Brodersen not only for chairing my qual committee but also establishing BWRC, teaching me circuit design and being such a fine role model to the rest of us.

What made my Berkeley experience so unique was the presence of multi disciplinary work in our lab as well as the friendliness and technical caliber of the fellow researchers around us. Ian O'Donnell has always been a great friend. He was not only my role model and mentor but his breadth of knowledge and experience was always inspiring. He has taught me so varied things ranging from effects of charge pump current mismatch to proper weight lifting techniques to drum and base music. Ada Poon is one of individuals whose friendship I most appreciate. Extremely smart, friendly, kind, thoughtful she definitely was one of the people that added a lot of meaning to my life in Berkeley. Mike Sheets was the person I worked most closely with. He is also one of the sharpest guys I know. During every project we worked

on at some point he made me utter the words "I'm impressed!!" may it be for his clock tree synthesis with eight clocks or golden make files or synthesis scripts. Mike Shuo Wei Chen definitely needs acknowledgement for his great friendship during the course of our times in BWRC.

Former BWRC members Peimin Chi, Kosta Sarigeorgidis, Josie Ammer, Sayf Alalusi, Stanley Wang, Brian Limketkai John Vanderhagen, Yun Chiu, EnYi Lin, were wonderful lab mates and great friends.. They were all very helpful and friendly and resourceful. They are all very special people. Current BWRC members Dave Sobel, DeLynn Bettencourt, Mubaraq Mishra, Hayden So, Dejan Markovic, Daniela Cabric and others were always fun to hang out with. Tom Boot and all the other BWRC staff is kindly acknowledged for their great help during my time at BWRC. Also the departmental staff, especially Ruth Gjerde, was always helpful and accommodating.

Outside BWRC, the members of the "Kultur sanat ekibi" namely Yalin Arici and Suzan Yilmaz were sources of endless support and friendship. I really cherished their friendships as well as the CalPerformances we caught together. In addition, I would like to acknowledge Hakan Dogan for his selfless friendship over my Berkeley years. Hasan Karatas and Safak Oguz were a great bunch guys who added tons of memories to reminisce in the years to come. In my last year of studies Cagla Meral and Seval Pinarbasi were very supportive friends and gracious hosts during many dinners. Also Elif Nur Firat, with her presence, was a continuous source of support and love in the last stretch of my PhD work .

My parents and my brother have always been my greatest supporters. I turned to them when in trouble, they consoled me when I felt down, they cheered hardest when I achieved... Therefore it is only fair that I dedicate all this work to them will all my heartfelt gratitude.

DARPA, NSF, GSRC were key funding organizations for the research carried out

in this thesis. Also the Turkish Ministry of Education is acknowledged for its funding during my undergraduate education at the University of Michigan.

Chapter 1

Introduction

> "It is He Who maketh the stars (as beacons) for you, that ye may guide
> yourselves, with their help, through the dark spaces of land and sea..."

> Holy Quran 6.97

1.1 Localization

Localization is defined as "assigning ... a definite locality". This definition can be interpreted as computing the coordinates of a position in a given coordinate system. The foremost general applications of localization are navigation and tracking. The main use for these applications include personnel or equipment transportation for military or civilian purposes. However availability of location and position information can also enable a huge swath of secondary applications. Possibilities include using the location information for customizing products, services; improving ways of data communications, realizing smart homes and offices, providing improved emergency responses etc.
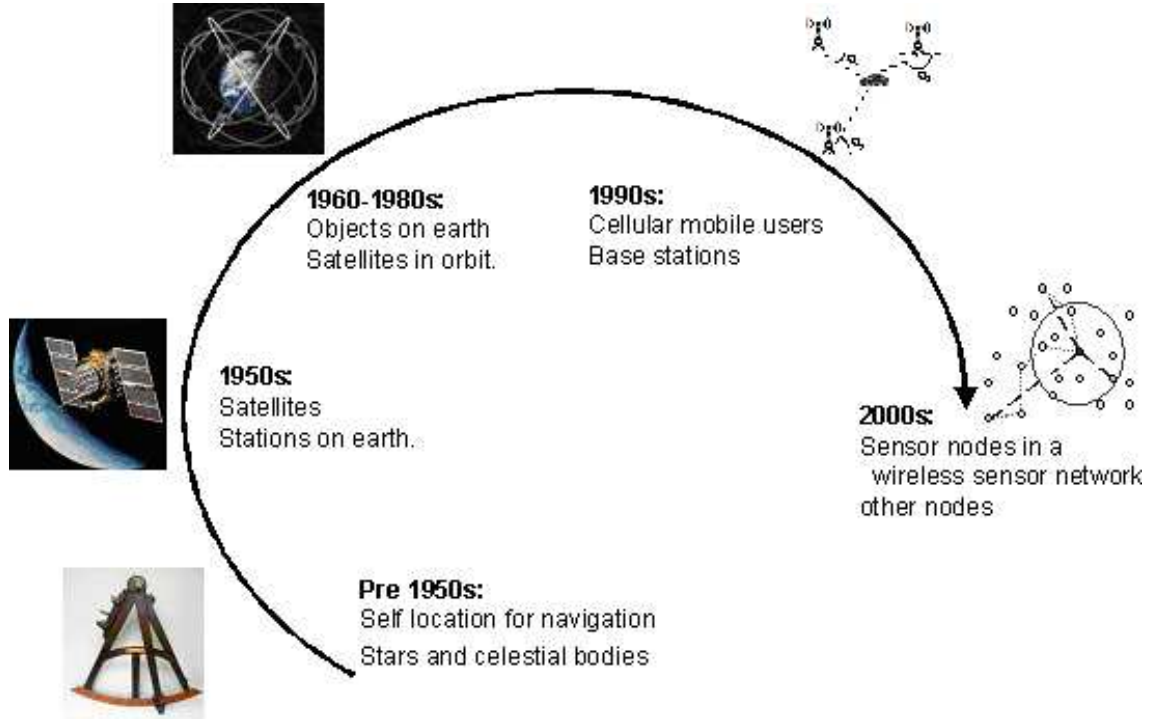
Figure 1.1: Localization time wheel

1.2 Historical perspective of Localization

Localization is an age old question that reappeared at many different contexts throughout history. It first appeared in the context of navigation when traveling through land and sea. Here the stars are used as reference points. The elevations angles of these stars from the horizon are measured using sextants. Using these two measurements the coordinates of unknown point is computed.

Until 1950s navigation was the main need for localization whereas use of stars were its main method. Then came up the need for tracking satellites in the space. This time the reference points on earth surface are used along with the distances between the satellite and earthbound reference points.

Rolling into operation in late 80s, Global Positioning System (GPS) was the next

significant localization application. Here satellites are used as the reference points and distances (or pseudoranges) between these satellites and the unknown point are measured. Next using the exact satellite locations, obtained from orbital ephemeris data, the pseudoranges the coordinates of the point to be localized are computed via matrix computations. [1]

Another emerging application since late 90s has been in locating mobile users in a cellular network during an emergency situation. In this application the cellular base stations serve as reference points. The angles or distances to these points are measured. Eventually using the base station coordinates and the distances or angles to these stations the mobile user position is computed.[2]

Finally in the 2000s another reincarnation of the localization problem emerged in the sensor network arena. In this space sensor nodes use their neighbors positions and the distances to these neighbors for computing their own positions.

1.2.1 Localization and its two subtasks

As can be seen in the historical progress of localization systems there is a common pattern that reemerges in many different localization applications. Once the reference points are available localization is completed in two steps:

1. Relate the unknown point to the reference points

2. Use reference points and relations to them to compute the final position algorithmically.

This process is also illustrated in Figure 1.2. In all these cases the operation starts with the availability or acquisition of a number of reference points. Next the unknown position is related to these reference points. These relations can be in different forms [3]. Often they are distances to the reference points, as in the case of GPS. However

1. Measure distance/connectivity/angle to relate to reference points

2. Compute location.

Reference point 1

Θ_1

r_1

Node being located

r_2 r_3

Θ_2 Θ_3

Reference point 2 Reference point 3

Reference point 1

r_1

Located node

r_3

r_2

Reference point 2 Reference point 3
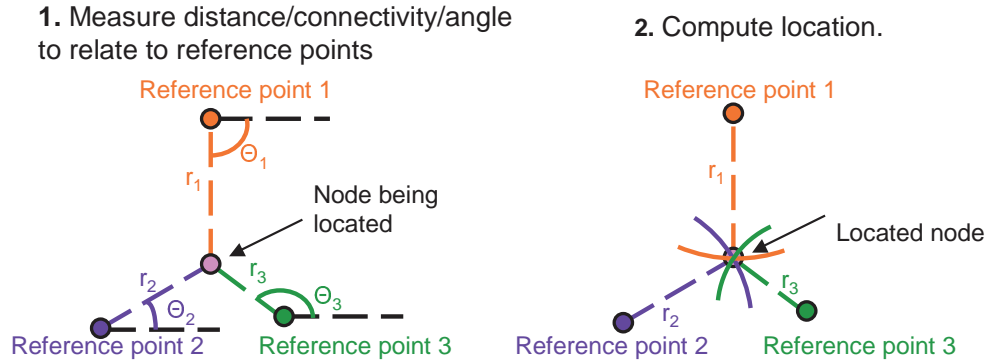
Figure 1.2: Two steps of localization

angles can also be used when relating the unknown point to the reference points. This is the case that when stars are used as reference points. Then sextants are used to measure their elevations from the horizon. In certain cases even the presence or absence of radio link between the reference point and the unknown point can be used to establish such a relation. [4, 5]. Once the reference points and the relations are established these information are used to compute the coordinates of the unknown position.

1.3 Sensor Networks

Ubiquitous, self-configuring sensor networks hold the potential of many new applications in monitoring and control. For example, climate control, intrusion detection, visitor guidance, and target tracking can be named as such. These networks are comprised of a large number of low power low data rate wireless sensor nodes. The nodes are deployed densely in a sensing environment such that their maximum physical separation would be around 10m [6]. Therefore the acquired sensor data can be used for different purposes from environment micro management to creating gradient maps of the sensor data. The second key characteristic of these networks is that the data

is usually acquired infrequently leading to low data rate communications and bursty network data traffic.

Due to the high number of sensor nodes in the environment, self configuration is highly critical. That is, even though a high number of nodes are needed, their deployment is kept manageable by having a self configuring network where many nodes establish their positions, IDs, connectivity, etc. after deployment. The final requirement of sensor networks is that to ensure operation of a high number of nodes over an acceptable length of time without intervention, the nodes need to be consuming very low power such that they can operate on a single battery for years or harvest energy from their surroundings. The target average power consumption of sensor network circuits is around 100μW [6].

In short, wireless sensor networks are characterized by high density of nodes and low speed communication requirements. As consequences of these two characteristics, mostly the first characteristic, sensor networks need to be self configuring and very low power consuming.

1.3.1 Localization in Sensor Networks

As the most recent reincarnation of the age old localization problem, localization in sensor networks has been attracting a large research effort in the last decade. Location information in sensor network setting can be useful for many different purposes. These include

- Acquiring location information during ad-hoc self reconfiguring deployment of wireless sensor networks.

- Associating sensor data with context information such that the sensor data would have physical meaning.

- Improving or enhancing the functions provided the sensor network such as improved routing, tracking capabilities.

In order to provide sensor nodes with position information a simple solution would entail programming each node at installation or adding GPS receivers in each sensor node. However these solutions would be first costly due to the high number of sensor nodes. Secondly they may simply not be appropriate, for example in the case of GPS receivers operation indoors is not possible. Therefore the more favorable solution is when the node positions, at least positions of most nodes, are computed after deployment. Therefore Localization is an important task for self-configuring wireless sensor networks.

There are a number of specifications imposed on the localization problem by the requirements of the sensor network properties. First is the accuracy requirement. An acceptable position error is around 0.5m [7]. Usually specified as a function of the maximum radio range, which is 10m in the sensor network case, this corresponds to a 5% of the radio range. Additionally a ranging errors up to 1m or 10% of the radio range are specified for the sensor network localization.

To achieve such a localization result an average power consumption lower than 100μW is required. This specification is determined by the 100μW average power budget foreseen for the total node power. This is usually achieved by activating the localization operation at a very low duty cycle. In addition, a maximum power consumption of 40mW is specified for the localization operation. This is a consequence of the energy storage capabilities of the node power train [6].

To keep the installation costs down as few nodes as possible need to be programmed during network installation. A maximum of 10% of all nodes are allowed to be preprogrammed with their positions. Finally the network is assumed to be mostly static that is nodes can be added to, removed from or move within the network slowly

in a matter of minutes. Therefore the proposed and implemented localization methods need to be satisfying these requirements from wireless sensor network properties.

1.4 Contributions

As mentioned in earlier localization in general consists of two tasks, establishing relations to the reference points and computing the positions using these relations. When we look at the existing research regarding sensor network localization, the first thing to notice is that bulk of the research has been focused on developing localization algorithms. Whereas implementing these algorithms as well as proposing and implementing ranging systems have not been pursued as aggressively. Therefore there has been a rich literature on localization algorithms, their first order performance analysis and simulations in comparison to implementations of ranging and position computation aspects of these localization systems.

In contrast in this thesis we would like to focus on the implementation issues relating to localization systems. These issues entail real hardware realizing such functions as well as their performances and costs. To this end we implement the two critical subtasks of localization. For the first task a triangulation (aka trilateration) algorithm is implemented in hardware. The design is implemented in a 0.13μ CMOS process. It dissipates 1.7mW of active power and 0.122nJ/op of active energy with a silicon area of 0.55mm^2. The mean calculated location error due to fixed-point implementation is shown to be 6%.

For the second part a ranging system is proposed and a prototype is implemented and its performance is evaluated in a real wireless channel. The prototype measurement error is within -0.5m to 2m while operating at 100Msps sampling rate and using a 50MHz signal in the 2.4GHz ISM band. The system accuracy is limited by the

sampling rate and can be linearly improved with increasing rates. This RF method is more cost effective than acoustic signal based ranging schemes, as it does not require ultrasonic transducers. The system is multipath resilient and can coexist with 2.4GHz band devices such as 802.11b/g networks. The estimated power consumption for the digital baseband is 2.35mW and its estimated area 0.25mm^2 in a 90nm CMOS process. The analog and RF sections have an estimated power dissipation of 38mW.

In summary, the main contribution of this work is to add these important implementation aspects regarding distance measurements as well as localization computations into the existing body of knowledge and bring it to the attention of the research community at large.

1.5 Thesis organization

The rest of this thesis will be organized as follows. Chapter 2 will begin with a survey of existing localization algorithms. Chapter 3 will define the triangulation method which is a key algorithm in the implemented localization system. Chapter 4 will describe the localization algorithm implementation and related results. Chapter 5 will be discussing Ranging in Sensor networks. Basic techniques are reviewed along with a comparative selection of the appropriate ranging method. Chapter 6 describes the implemented ranging system as well as presenting detailed discussions on selecting design parameters. Chapter 7 details the digital baseband implementation of the ranging system along with some background. Chapter 8 shows the prototype of the proposed ranging system as well as presenting some real data. The thesis complete with a conclusion and an outlook considering possible extensions and improvements on the performed work.

Chapter 2

Sensor network localization

As discussed in the first chapter location information can have many uses in sensor networks. However due to the high number of nodes in these networks, it is highly desirable that a small fraction of nodes are preprogrammed with their positions while most of the sensor nodes independently compute their own positions after deployment. Hence only a few numbers of nodes would be given a priori information about their positions with respect to a global coordinate system. These nodes are called anchor nodes or beacon nodes. The rest of the nodes then calculate their positions and localize themselves by using positions of the anchors and their own relations with these anchors.

There are several challenges for the localization endeavor in sensor networks. First, a solution has to be tolerant to large errors in (e.g. range) measurements. Second, the complexity of the localization algorithm must not grow faster than the network size. Third, the algorithm should operate in awareness of the constrained communication and computation resources in a sensor network setting [8]. That is it should consume low power and energy as well as requiring minimal communication.

Among the research alternatives on sensor network localization, studies regard-

ing different localization algorithms are most common. That is a large number of localization algorithms have been proposed in literature. Moreover, there has been a few attempts at classifying these methods proposed. For example in one source [9] the localization algorithms have been classified into active and passive as well as cooperative target and cooperative infrastructure based methods. In this work the localization algorithms are classified in terms of the locations where computations take place. That is at a central location or at every node. Therefore the considered classes for localization algorithms are centralized and distributed.

With this centralized and distributed classification of algorithms sensor networks, with their fully homogenous structure, at first seem inherently more appropriate for distributed localization approaches. However when node differentiation is allowed, which means some nodes can have more resources and carry heavier computational loads, the ideas from centralized localization approaches become more relevant and applicable. Hence both centralized and distributed localization approaches bear practical importance for various scenarios.

As stated earlier there exists a rich body of literature regarding sensor network localization algorithms for many different applications, specifications and performances. Despite implementation-specific constraints of a sensor network and the diverse requirements of each application running on the network, a set of global performance criteria can be devised. For specific applications, only a subset of the criteria may be relevant. The following criteria encompass both necessary and desirable properties of any localization algorithm [8]:

- **Accuracy**. Some applications may require an upper bound on the estimation error.

- **Sparse anchor tolerance**. Even with very few anchors the system should be able to function and localize the nodes without initial position information.

This would typically lower the deployment complexity and cost.

- **Error tolerance**. The localization algorithm must work with range measurement errors. Range errors occur because of the finite Signal to Noise Ratio SNR of the received signals to perform ranging or any kind of relationship. Also sampling effects can induce measurement errors.

- **Scalability**. A scalable algorithm keeps the required per-node computation constant as the network size grows. This property is critical to be able to support networks of different sizes without redesigning the localization system.

- **Energy dissipation**. It is desirable to minimize total computation energy in the network. However, there is often a trade-off between computation energy, speed of convergence, and communication requirements. in addition it is also desirable to keep the total energy spent on communication as low as possible.

- **Convergence time**. Applications may need fast convergence times, for example becomes critical if a mobile node, which can be attached to a human, localizes itself.

Once these criteria are established for comparing various methods of localization, each significant algorithm in literature should be visited. First group that will be reviewed include centralized algorithms followed by a study of distributed approaches. Our review in the rest of the sections is based on the sensor network localization reviews prepared by Van Greunen [8] and Savvides et.al [9].

2.1   Centralized computations

Centralized computations have the common trait where data is collected to a single unit and the location of all the nodes are computed all at once within this unit.

The most common issue raising from centralized computations is the communication overhead for bringing the data to the computation node. Moreover distributing the computed positions from the computation node to the rest of the nodes in the network also adds communication overhead. Also very often in centralized algorithms computation complexity for the processing node grow with a more than linear relationship. That is increasing the network size increases the load on the processing node more than the load increase in a distributed computation node.

2.1.1   Centralized Linear Programming

The first centralized localization algorithm considered in this review uses radio connectivity as a relation between reference nodes and sensors [4]. Main assumption in this method is that if two nodes can communicate with each other then they are within a circular radio range (R) from each other.In other words, if two nodes can communicate one node is located within a circle centered at the other node. This assumption can also be mathematically formulated as

$$||a - b|| < R \qquad (2.1)$$

where a and b are the position vectors of the two nodes. Next the constraints from each node can be brought together to obtain a global problem formulation as

$$\text{Minimize:} \quad c^T p$$
$$\text{subject to:} \quad Ap < b \qquad (2.2)$$

Where p is a matrix obtained by using (x, y) coordinates for each node in the network. The Linear Programming (LP) problem needs to be solved for each unknown node in the network; to find four the corners of the feasible bounding box around the

unknown node, c_{2k} or c_{2k-1} is set to 1 or -1 while all other entries are 0. If the objective function, $c^T p$, is omitted, it has the same effect as selecting a random point within the bounding box. To find the constraint, $Ap < b$, Schur complements can be used . A standard LP solver can then be used solve this convex problem [8, 4].

This method offers many advantages for sensor node localization. first and biggest advantage is that it does not need any other relation information but connectivity. Therefore any ranging or received signal strength measurements not necessary but still useful improved performance . Additionally by formulating the task as a convex optimization problem the rich body of existing knowledge in this subject can be exploited. Also the algorithm is quite tolerant to measurement noises since the only measurements are binary decisions.

The key disadvantage however is that the computation needs to be executed centrally. Therefore the connectivity of each node needs to be communicated to a central location. This creates extra network traffic, possibly bottlenecks, and computational load on computation nodes. Also the quadratic dependency of network size to computational complexity makes the algorithm unsuitable for scaling. Moreover even though the radio coverage is expected to be circular in fact the radio range varies with direction as the channel indoor effects play out. As a final note the settling time of the algorithm depends on the network size and the end to end distance of the network.

2.1.2 Kernel Based learning localization

This method uses connectivity of sensors and kernel based regressions as well as classifications to perform localization [5]. The anchor nodes are used in kernel based learning and a classification based on these anchors is created. Next, using the connectivity of the unknown nodes and the classification obtained from anchor nodes,

the unknown node is localized. In a way this is similar to modeling the localization problem and then using this model and node connectivity in estimating the positions. Hyper planes are used to separate the space into possibly overlapping regions and using connectivity of nodes they can be associated with these regions. The more anchors there are the smaller the regions would be and the finer the location estimate would get.

When the network size is $L \times L$, radio range is R and the number of anchors is A the position error would be in the order of $O(A^{-1/6}L^{1/3}R^{2/3})$. That is the algorithm has a sense of its error magnitude.

The accuracy of the algorithm is somewhat crude and but it can be improved by using more anchors. Also being a statistical algorithm, it is sensitive about not having enough anchors to begin with. Finally being a central localization algorithm it is not very scalable.

2.1.3   Radio signal strength database

Next considered method is another centralized search algorithm [10]. Signal transmitted from an unknown position is received at multiple receivers. Using the Received Signal Strength values as a vector and comparing them against a database, where each RSS vector is associated with a certain position, the unknown position is estimated. I.e. the database entry that is closest to the received vector is selected as the location estimate.

In a way this algorithm is a less sophisticated version of the kernel based learning algorithm. While the RSS DB algorithm performs a brute force database search for the search for the optimal point, the kernel based method tries to extract kernels from that database and performs the comparisons using these kernels which "span" the database entries. The biggest advantage of this method is that it is conceptually

extremely simple, quick for deployment and has been used in some real life situations. Also the algorithm only needs the RSS measurements which are quite cheap to obtain. The search algorithm can be simple but can be time consuming depending on the search space size. Finally the algorithm is robust against indoor channel effects.

The biggest challenge of this algorithm is its large installation cost. For getting any accuracy, there needs to be a large number of position entries in the database. Also having central computations this algorithm is not very scalable, the problem size grows as the number of points grows. The need to store a database of reference points imposes heavy storage requirements which can create inconveniences during implementation.

2.2 Distributed algorithms

Distributed algorithms are very often used in sensor network localization problems. Indeed due to the inherent distributed structure of sensor networks a distributed solutions is more often than not deemed more appropriate for sensor network localization. What's more this problem is often labeled as a distributed optimization. One general problem with the distributed algorithms approach is that convergence of the results can take long which maybe inappropriate for some applications.

2.2.1 Rectangular intersection

Rectangular intersection is also based on the idea of connectivity [11]. Similar to the Centralized LP algorithm it assumes two communicating nodes have implications regarding their positions. However here if two nodes can communicate with each other, it is assumed that one is within a square centered at the other node and with a side length equal to twice the radio range.

The main advantage of this algorithm is that intersecting squares is a mathematically simpler operation than intersecting circles. The reason is when intersecting squares intersection is a rectangle however when intersecting circles the intersection is much harder to describe mathematically. Each rectangle can be represented by its upper right and lower left corners. If all the neighboring nodes have the their centers at coordinates x_i, y_i then the corners of the intersection rectangle can be defined by

$$
\begin{aligned}
[x_1, y_1] &= \left[\max_{i \in 1,...,N} \{x_i\} - R, \max_{i \in 1,...,N} \{y_i\} - R \right] \\
[x_2, y_2] &= \left[\min_{i \in 1,...,N} \{x_i\} + R, \min_{i \in 1,...,N} \{y_i\} + R \right]
\end{aligned}
\tag{2.3}
$$

The second advantage of this algorithm is that it is executed in a distributed fashion.At each node one, positions of neighbors are acquired. Then squares centered at these neighbors are intersected yielding a final rectangle. The final position estimate is the center of the intersection rectangle. The disadvantages of this method is the dependence on convexity of communication regions as well as the need for high connectivity to obtain a refined and accurate estimate.

2.2.2   Grid of beacons

In this algorithm the anchors are placed on a rectangular grid. [12, 13] The node to be placed first determines which nodes it can communicate with then finally it computes its own position at the centroid position of the anchors it can communicate. That is if the node can communicate with a total of M anchors with indices i_k, then the estimated position is at

$$
[\overline{x}, \overline{y}] = \left[\frac{\sum_{k=1}^{M} x_{i_k}}{M}, \frac{\sum_{k=1}^{M} y_{i_k}}{M} \right]
\tag{2.4}
$$

The main advantages of this approach are again the complexity and the implementation simplicity of the approach. The main disadvantage is the costly setup and

initialization of this network. There is need for a grid of anchor placement whether or not it is possible. The research indicated that there is an optimum mesh that the nodes can be placed to optimize the position error [13]. Other than that the position accuracy degrades.

2.2.3 Triangulation

The simple idea behind triangulation method (a.k.a. trilateration) is that if the distance (d) between two nodes is known, then one node must be on a circle centered at the other node and with radius d [14, 15]. Note that in this algorithm the relation to the reference nodes are no longer connectivity but it is the distance between the node and the reference node.

When distances to multiple nodes are known. The circles centered at these reference nodes with radii equal to the distances are intersected and at the intersection of these circles lies the point with unknown position. This scheme will be illustrated in Figure 3.1.

The main advantage of triangulation scheme is that, to first order, its functionality does not really depend on the anchor density. However when there are range errors the solution does have an error. In the case with noisy range measurements, having more reference points than necessary can create an averaging effect as the range error in each measurement would have varying errors. Such a solution could be implemented using optimization techniques such as Least Squares optimization.

Another important consideration is the accuracy of the ranging measurements. It is obvious that the less noisy the range measurements are the more accurate the final position estimation would be. As will be seen later in this work performing accurate range measurements in the sensor network setting is a challenge in itself. However, only by using range measurements, localization with acceptable performance can be

achieved without consorting to expensive infrastructures such as beacon grids or RSS calibration databases.

2.2.4   DV Hop - Hop Terrain

DV-Hop or Hop-Terrain is a method that shares great similarities with the triangulation algorithm [16, 17, 18]. Here the main difference is that instead of the measurements of the real Euclidean distance between nodes an average distance is used. This average distance utilized in this case is the number of hops from an anchor node in the network. To convert the number of hops to a real distance an average distance for a hop can be used.

To compute the hop distances the anchor nodes act as the reference nodes of the network and flood the network with their positions. Remaining nodes maintain a list of these reference points and the number of network hops it takes packets to reach it from these anchor nodes also known as "hop-count or hop-distance". Once a node knows adequate number of reference numbers and their associated hop distances it performs a triangulation to compute their positions.

The advantage of this method is that it achieves a localization with a rather simple range measurement technique. Additionally, the algorithm is scalable since the nodes need to hear from only four reference points to be able to compute their own positions. On the other hand, due to using crude distances the estimates are quite noisy and inaccurate. The position errors can go up to the order of the radio range [19]. The algorithm can perform especially worse when the network topology is very irregular and actual distances are much different from the real distances. To reduce the associated position errors using a larger number of anchors and a smaller average hop distance is beneficial. That is, using more nodes as well as more anchor nodes can improve the position estimates.

2.2.5   Coordinate rotation

Another localization algorithm that utilizes the distance measurements is the co-ordinate rotation algorithm [15, 20]. This algorithm also uses triangulation for local-ization operations. The novelty of this algorithm come from the fact that here there is neither any predetermined coordinate system nor any nodes programmed in such a system. Instead every node computes itself a position by assigning itself a random coordinate system. Usually that proceeds by assigning one neighbor on the x-axis the next neighbor on the xy-plane, so on and so forth. Then the arbitrary coordinate sys-tems are rotated until the different coordinate systems across the network are aligned and one single coordinate system is achieved all throughout the network.

The main appeal of this algorithm is the lack of anchor nodes and the need for any installation efforts. The main drawback of this algorithm is the accumulating range errors during the coordinate rotations.

2.2.6   Multilateration

The multilateration algorithm is the first algorithm that we consider which com-putes the location in stages [21]. In essence it is a hybrid algorithm taking advantage of different aspects of different methods. It consists of three distinct phases.

In the first phase, all ill-connected nodes are removed and the well-connected nodes are placed in collaborative sub-trees. Well-connected nodes are defined as the nodes that satisfy the following properties. Having more than three neighbors, having non-collinear reference points and finally for twin nodes, as defined by Savarese [19], having connections to other nodes that are not connected to the twin.

During the second phase, the nodes obtain initial position estimates. This stage borrows ideas from rectangular intersections method described earlier in this chap-

ter. The idea involves defining bounding boxes. However this time the measured distances and the distance measurement errors are used instead of the radio ranges in rectangular intersections method. When this phase of the algorithm completes every node would have an initial position so that they can perform an iterative refinement.

The last phase in the multilateration algorithm is the refinement phase. During this stage each node measures distances to its neighbors and computes a recursive Kalman filter operation. This is an approximation of a fully decentralized Kalman filter because the nodes do not exchange covariance information [8, 21].

This algorithm achieves reasonable accuracy provided good initial estimates. Therefore the first two phases are important to yield a good initial estimate. In order to obtain good estimates in the first phase the network needs to be as close to a regular mesh as possible. Excessive irregularities in the network topology would cause trimming of a high number of nodes. Moreover the pruning of nodes during the algorithm causes some nodes to be dropped from the localization process. For improving the second phase, using highly connected nodes can increase the accuracy of the initial estimate as well.

The Kalman filter in the refinement phase is a substitute for the triangulation algorithm. The essential difference from triangulation is that it can operate without the need for a minimum number of nodes. However computationally Kalman filter computations are rather complicated operations. Kalman filter gain, covariance matrix, measurement noise matrix, and Jacobian of the blending matrix must be calculated in order to realize the Kalman filter required for this approach. Moreover the iterative nature of the refinement phase can lead to instabilities and convergence problems. Finally one overall observation about the inefficiency of the multilateration algorithm is that none of the functions from any of the phases can be shared leading to extra hardware requirements.

2.3   Selected approach: Two Phase Triangulation

The localization approach selected for implementation is also a hybrid method that offers a compromise between different design specifications. This method is known as two phase triangulation or start-up and refine algorithm. At the core this algorithm computes triangulation for localization.

At startup not all nodes are in the radio range of enough number of anchor nodes. In order to achieve an initial position estimate these anchors are used as reference points and the number of hops to these anchor nodes are used instead of real distances. Hence, the described above triangulation uses hop counts instead. This algorithm was already described above and was initially proposed by Savarese et.al. [19]. Independently Niculescu et.al also suggested a similar algorithm [16].

Once each node acquires an initial position. They in turn start using their immediate neighbors as immediate points. By measuring the Euclidean distance to these neighbors and using neighbors' initial positions nodes perform a secondary triangulation to update their position. Simultaneously their neighbors perform triangulations and update their own positions iteratively.

The two phase localization algorithm is illustrated in Figure 2.1. Here it is clear that for both phases of the algorithm the main localization technique is triangulation. The only differences are the reference points and the distances used in the triangulation. In the first phase the reference points are the anchor nodes and used distances are the hop counts to these anchors. In the second phase the reference points are the immediate neighbor nodes and the distances are the real Euclidean distances.

The second phase of the algorithm repeats until the location estimate of the node converges to a value. However the iterative nature of the refinement stage causes concerns regarding the stability of the final location estimate. For this reason, additional features to ensure stability are added to the refinement phase.
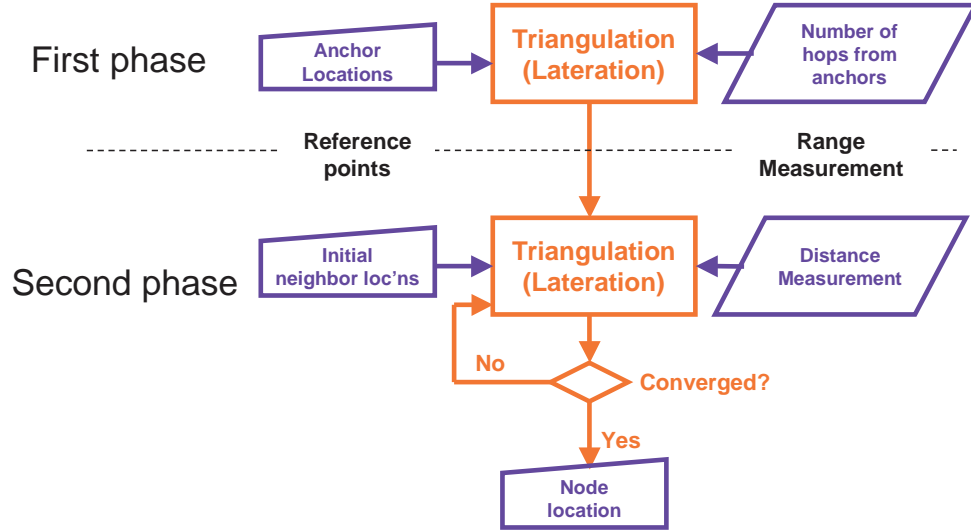
Figure 2.1: Block diagram of the selected Two-phase localization algorithm

The first measure is preventing ill connected nodes participating in the refinement. An ill connected node is a node that does not have independent neighbors. That is it has less than three neighbors who are not connected to each other. The advantage of such a pruning right before refinement is that even nodes which are denied to participate in refinement have their initial position estimates. Therefore they can participate in basic network functionalities. This is in contrast with the multilateration approach where the sick topologies are trimmed in the beginning, leaving those nodes without any position.

The second measure involves introducing a confidence metric such that the triangulation has a weighting effect from nodes which are more certain of their positions. Nodes with unknown positions assume an initial confidence of 0.1, whereas the anchor nodes assume a confidence of 1. When neighbors transmit their positions they also transmit their confidences and this information is used in the triangulation operations when updating the node location. During these updates node confidence is

also recalculated. As the algorithm progresses the confidence of the nodes localizing themselves start increasing from the vicinity of the anchors and propagate into the network. The iterations are terminated when the node confidence is settled to a value and does not change for many iterations.

The main appeal of the two phase localization algorithm is its accuracy. This is mostly due to the use of distance measurements as relations to reference points and use of over determinism for mitigating range error effects. In terms of accuracy only other algorithm that can measure up to the two phase algorithm is the multilateration. For the other algorithms to achieve comparable performance with the two phase localization or multilateration they would need to have higher setup or infrastructure costs. This is mostly due to the need for a higher number of anchor nodes required to improve estimates for these algorithms.

In terms of implementation complexity that both the two phase localization and the multilateration algorithms incur more complexity than the rest of the localization algorithms covered in this chapter. However a comparison between these two reveal some differences in terms of implementation. The first difference is the refinement algorithm. The triangulation algorithm is less complex than Kalman filtering [8]. The second difference is that multilateration performs connection conditioning and node pruning before it even starts localization whereas two phase localization performs this pruning after initial position assignments. This way, no node is left behind without position and each node can participate in network functions without any difficulty. The final advantage of two phase localization is the reusability of the solutions in different phases. For instance the triangulation unit can be used in both the start-up and refinement phases. What's more when real distance measurements are not available the algorithm can still initialize the network and have some crude localization information available. However in the multilateration algorithm even for startup if

range measurements are not available the algorithm can not proceed. That is this algorithm is less flexible than two phase localization. Moreover the solution hardware and software are not reusable across phases. Because of its accuracy and these above mentioned implementation reasons the two phase localization was the selection of implementation.

2.4   Conclusion

In this chapter a background review on different localization algorithms are presented. The algorithms are divided into two classes centralized and distributed localization algorithms. After a detailed review of different algorithms a distributed two phase localization algorithm was selected for implementation. This algorithm combined advantages of ranging based methods as well as connectivity based methods to achieve accuracy and robustness of the algorithm. Next will be presented is aa hardware implementation of this algorithm.

Chapter 3

Triangulation system design

It was discussed in the previous sections that one significant task of localization is computing the unknown location given a number of reference points and relations to these reference points in the forms of distances. Even though this task is carried out as the final task of localization it will be studied here as the first task.

Also in the previous section after going through the alternative sensor network algorithms. The two phase localization algorithm, which consisted of the two phases of Hop-Distance-based initialization and Euclidean distance based refinement, was chosen for implementation. This algorithm uses the distance of the unknown point from the reference points to compute the distances.

3.1 Problem formulation

The locus of a point that is at a distance d away from a reference point is a circle (in a two dimensional or 2-D space) or a sphere (in a three dimensional or 3-D space) of radius d centered at the reference point. In simple terms, when distance (d) from a reference point is determined, this implies that the unknown point lies on a

circle or a sphere with a radius of d and centered at the reference point. Moreover, the unknown point is known to lie on a number of circles simultaneously, each of which are centered at one reference point and have the radius of the distance to that reference point. Therefore the overall effect is that the unknown point lies in their intersection point. To obtain a unique point, and resolve any ambiguities, three (in 2-D) or four (in 3-D) reference points are needed [15]. This method of intersecting the circles is referred as *Triangulation (or Trilateration)* and is illustrated in Figure 3.1.

In Figure 3.1 assumed is a 2-D space. The locus due to the distance r_1 to Reference Node 1 is a circle centered at that point and a radius of r_1, next the loci due to the Reference Nodes 1 and 2 are the intersection of the two circles centered at these points. As can be seen in the figure this intersection of two circles consists of two points. To resolve this ambiguity a final reference point is needed. Using this Reference Node 3 point a unique coordinate can be computed for the node being located. As a general principle in order to perform a triangulation in an N-dimensional space N+1 reference points and the distances to these points are needed.

However to compute unknown positions mathematically, the operations in Figure 3.1 need to be formulated mathematically. For this purpose, the equation for finding the distance between two points can be used. Here, assuming a 3-D space, suppose (u_x, u_y, u_z) are the coordinates of the unknown position and (x_i, y_i, z_i) are the coordinates of the ith reference point for $i = 1, \ldots, n$. Then, the coordinates and the distances are related by the set of equations

$$
\begin{bmatrix} (x_1 - u_x)^2 + (y_1 - u_y)^2 + (z_1 - u_z)^2 \\ \vdots \\ (x_n - u_x)^2 + (y_n - u_y)^2 + (z_n - u_z)^2 \end{bmatrix} = \begin{bmatrix} r_1^2 \\ \vdots \\ r_n^2 \end{bmatrix} \tag{3.1}
$$

where r_i is the distance between the i_{th} reference point and the point with unknown
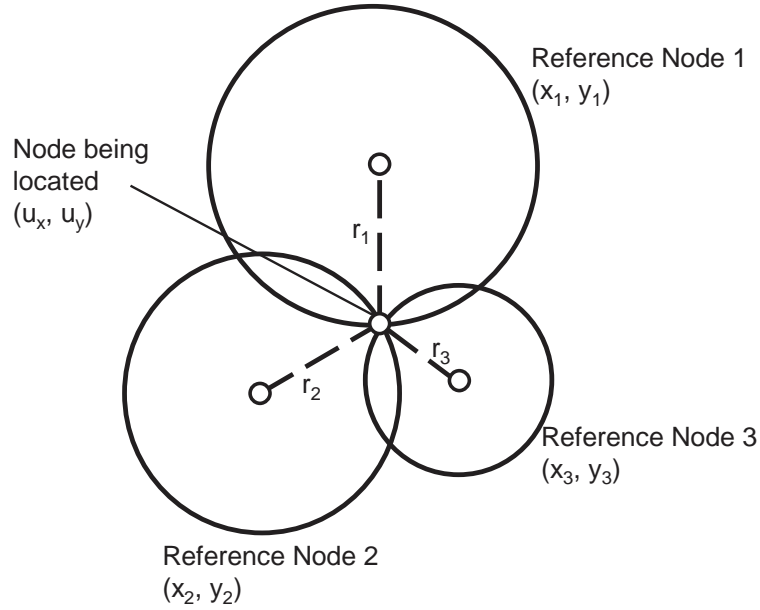
Figure 3.1: Illustration of triangulation in 2-D. Three loci are intersected to yield the unknown position

coordinates.

These nonlinear equations relating the unknown coordinates to the distances and reference point coordinates can be solved in a number of ways. This can involve using either closed form solutions [1], iterative techniques based on linearization or Kalman filtering [1, 21]. The biggest advantage of using closed form solutions is that it can operate without any assumption for an initial value for the unknown position. Nevertheless this comes at the expense of more sophisticated computations and a requirement for a minimum number of reference points. Linearization and Kalman filtering on the other hand assume initial values for the user position and compute the difference from that assumption. Hence they can operate using simpler computations and fever number of reference points. However they assume availability and reliability of initial position estimates.

Expecting initial estimates needs these estimates from either being provided dur-

ing setup or running an initial localization with a closed form solution to Equation 3.1. Providing these initial estimates at setup would increase system deployment complexity, which is one property the whole localization operation intends to simplify. Providing them with an initial closed form solution would increase the system hardware complexity.

The two phase localization algorithm, which is selected for implementation in the previous section, outright solves the nonlinear equations in Equation 3.1 in closed form. After the initial solution this method is maintained for the subsequent localization computations and therefore the state and therefore "memory" of these computations are minimized.

To reach the closed form solution, one can start by subtracting the first line in Equation 3.1 from each of the remaining equations. This yields a linear system of $n-1$ equations, which can be written as

$$Au = b \qquad (3.2)$$

where,

$$A = \begin{bmatrix} (x_1 - x_2) & (y_1 - y_2) & (z_1 - z_2) \\ \vdots & \vdots & \vdots \\ (x_1 - x_n) & (y_1 - y_n) & (z_1 - z_n) \end{bmatrix} \qquad (3.3)$$

$$u = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \qquad (3.4)$$

$$b = 0.5 \begin{bmatrix} r_2^2 - r_1^2 - x_2^2 + x_1^2 - y_2^2 + y_1^2 - z_2^2 + z_1^2 \\ \vdots \\ r_n^2 - r_1^2 - x_n^2 + x_1^2 - y_n^2 + y_1^2 - z_n^2 + z_1^2 \end{bmatrix} \qquad (3.5)$$

This system of equations is over-determined if $n > 4$. For the case *if there are no ranging errors* any four of the $n - 1$ total equations can be selected and used for the triangulation operation. Again assuming no errors on the range measurements using 4 or $n - 1$ equations would not yield any additional information. In this case the unknown position can be computed using

$$\mathbf{u} = \mathbf{A_4^{-1} b_4}$$ (3.6)

where A_4 and b_4 are any four rows of A *and* b

However *when range measurements are noisy* and $n > 4$ then selecting any 4 equations would not all yield the same solution. In this case the availability of the additional reference points can be used to the advantage [15, 17, 14]. The overdetermined system of equations can be solved using a least squares solution. This is a technique borrowed from linear algebra that is often used in applications that consists of overdetermined systems with noisy measurements. Moreover when there are a few nodes with higher range errors that the average error, least squares (LS) optimization technique improves the final position calculation with comparison to the case when only 4 equations are selected and selection includes measurements with higher ranging error than average [14]. In which case the unknown position vector u, which is the solution to the vector system in Equation 3.2 can be determined as the least squares solution to this system and be formulated as

$$u = (A^T A)^{-1} A^T b$$ (3.7)

after the canonical closed form LS solution presented in [22]. Also it can be added that the vector $A_p = (A^T A)^{-1} A^T$ is sometimes called the pseudo inverse of A such that $u = A_p b$.

3.2 Least-squares Solution

Least squares (LS) solution is a method often used to determine the solution to an overdetermined system of equations with noisy observations. The system is overdetermined when there are more equations than unknowns. The implication of overdetermined matrices on the data matrix A is that it will have more rows than columns, i.e. $m > n$ is A is an $m \times n$ matrix. It should be noted that even though LS is a technique often used in linear algebra and matrix contexts, the same concepts also find use in estimation theory as Linear Least Squares techniques.

For a matrix equation system such as $Au = b$ the optimization procedure strives to find an u vector to minimize the norm $||Ax - b||$ [23]. The problem can be formulated as

$$\min_{u \in \mathbb{R}} ||Au - b|| \tag{3.8}$$

To derive the LS solution which was stated previously in Equation 3.7 Golub and Van Loan can be utilized [23]. Suppose $x \in \mathbb{R}^n, z \in \mathbb{R}^n,$ and $\alpha \in \mathbb{R}$ and consider the equality

$$||A(x + \alpha z) - b||^2 = ||Ax - b||^2 + 2\alpha z^T A^T (Ax - b) + \alpha^2 ||Az||^2$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. If x solves the LS problem in Equation 3.8 then we must have $A^T(Ax - b) = 0$. Otherwise, if $z = -A^T(Ax - b)$ and α is small enough it is possible to obtain a contradictory inequality $||A(x + \alpha z) - b|| < ||Ax - b||$ since x was already assumed to minimize $||Ax - b||$. Thus if A has full column rank,then there is a unique LS solution x_{LS} and it solves the symmetric positive definite linear system

$$A^T A x_{LS} = A^T b \tag{3.9}$$

$$x_{LS} = (A^T A)^{-1} A^T b \tag{3.10}$$

Methods to compute the least-squares solution of an over-determined linear system have been extensively studied and well published [22, 23, 24]. These methods can first be classified into two, the first being the use of Normal Equations and the second being use of QR factorization.

- Method utilizing normal equations start from Equation 3.9. After renaming $C = A^T A$ and $d = A^T b$ the Cholesky factorization for C is computed as $C = GG^T$. Finally $Gy = d$ and $G^T x_{LS} = y$ is solved to reach the LS solution.[23]

- For the method utilizing QR decomposition, let $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ and $b \in \mathbb{R}^m$ be given and suppose that an orthogonal matrix $Q \in \mathbb{R}^{m \times m}$ has been computed such that

$$Q^T A = R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \begin{matrix} n \\ m-n \end{matrix}$$

  is upper triangular. Furthermore assume

$$Q^T b = \begin{bmatrix} c \\ d \end{bmatrix} \begin{matrix} n \\ m-n \end{matrix}$$

  then it can be written that

$$||Ax - b||^2 = ||Q^T Ax - Q^T b||^2 = ||R_1 x - c||^2 + ||d||^2 \quad (3.11)$$

  for any $x \in \mathbb{R}^n$. Clearly the error norm $||Ax - b||$ is minimized when $||R_1 x - c|| = 0$ or $R_1 x_{LS} = c$. Also noteworthy is that $||Ax_{LS} - b|| = ||d||$ [23]

Both of the above methods and their respective derivations assumed a full rank matrix. When the matrix A is rank-deficient then there are methods such as pivoting that can be employed and studied in detail in literature. [22, 23]

When comparing these two methods for implementation. Again the discussion in Golub and Van Loan [23] proves useful. Quoting from this reference when $||Ax_{LS} - b||$

	Householder Reflections	Givens Rotations
Description	•Zero one column at a time. •Compute norm of a vector and divide by norm.	•Zero one entry at a time. •Measure angle of a complex number. Rotate a row by that angle.
Advantage	Less op's: 2n²m-2/3n² flops	Simple op: CORDIC
Disadvantage	Complicated ops: Division, square root	More op's: 4n²m-4/3n² flops

Table 3.1: Comparison of Householder reflections vs. Givens rotations

is small usually the method of Normal equations will produce a less accurate result than a stable QR approach. On the other hand the two methods produce comparably inaccurate results when applied to large residual, ill-conditioned problems. [23]

Finally the normal equations involve half of the arithmetic operations than QR method when $m \gg n$. However on the other hand not all arithmetic operations are as easy to implement and QR decomposition, as will be discussed soon, can be implemented with operations that are very easy to implement in hardware. Moreover QR approaches are applicable to a wider class of matrices because the Cholesky process applied to $A^T A$ breaks down "before" the back substitution process on $Q^T A = R$. [23]. Therefore due to accuracy, stability and implementation advantages the QR decomposition method is the selected LS solution approach.

In general, QR decomposition is applied to obtain an upper triangular matrix where back substitution with this triangular matrix yields the LS solution. For the QR decomposition, there are two standard algorithms: Householder reflections and Givens rotations.

- Algorithm using Householder reflections is an iterative approach. Householder reflections are modifications of the identity and they can be used to zero selected components of a vector. In a nutshell starting from a vector x a vector $v =$

$$
\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \\ \vdots & \vdots & \vdots \\ x & x & x \end{bmatrix} \mathbf{u} = \begin{bmatrix} y \\ y \\ y \\ y \\ \vdots \\ y \end{bmatrix} \xrightarrow{1 \to i} \begin{bmatrix} x & x & x \\ x & x & x \\ \tilde{x} & \tilde{x} & \tilde{x} \\ 0 & \tilde{x} & \tilde{x} \\ \vdots & \vdots & \vdots \\ 0 & \tilde{x} & \tilde{x} \end{bmatrix} \mathbf{u} = \begin{bmatrix} y \\ y \\ \tilde{y} \\ \tilde{y} \\ \vdots \\ \tilde{y} \end{bmatrix} \xrightarrow{i \to i+1} \begin{bmatrix} x & x & x \\ \hat{x} & \hat{x} & \hat{x} \\ 0 & \hat{x} & \hat{x} \\ 0 & \tilde{x} & \tilde{x} \\ \vdots & \vdots & \vdots \\ 0 & \tilde{x} & \tilde{x} \end{bmatrix} \mathbf{u} = \begin{bmatrix} y \\ y \\ \hat{y} \\ \hat{y} \\ \vdots \\ \tilde{y} \end{bmatrix} \xrightarrow{i+1 \to N}
$$

$$
\begin{bmatrix} \bar{x} & \bar{x} & \bar{x} \\ 0 & \bar{x} & \bar{x} \\ 0 & 0 & \bar{x} \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix} \mathbf{u} = \begin{bmatrix} \bar{y} \\ \bar{y} \\ \bar{y} \\ \bar{y} \\ \vdots \\ \bar{y} \end{bmatrix}
$$

Figure 3.2: Progress of QR decomposition using Givens rotations.The x and y represent non zero entries.

$x \pm ||x||e_1$ is constructed, where e_1 is the unit vector along the first dimension of the vector space. Next the matrix product

$$Px = \left(I - 2\frac{vv^T}{v^Tv}\right)x = \mp||x||e_1 \qquad (3.12)$$

is computed [23]. Matrix P is called a Householder reflection and is an orthonormal matrix. Applying these such transformations iteratively for the subsequent columns of the matrix A one can attain an upper triangular matrix R. along with an orthonormal matrix which is the product of all the Householder reflections.

It zeros out the lower part of one column of the matrix at each iteration. For an m-by-n matrix, its cost is $2n^2m - 2/3n^2$ floating point(FP) operations (flops) [22]. However, this operation count does not take into account the hardware complexity of implementing each flop. During each iteration, as seen in Equation 3.12 Householder reflections require computing the squared norm of an m-dimensional vector and a division by this squared norm. These operations are usually deemed expensive for hardware implementation.

- Algorithm using Givens rotations is also an iterative method. Givens rotations are rank-2 modifications to the identity matrix. This effect is apparent in Equation 3.13. When applied to a vector these rotations can rotate a complex number composed of the ith and k^{th} entry of the matrix. That is multiplying a vector $x \in \mathbb{R}^n$ and $G(i, k, \theta)$ the resulting vector y can be written as

$$y_j = \begin{cases} cx_i + sx_k & j = i \\ -sx_i + cx_k & j = k \\ x_j & j \neq i, k \end{cases}$$

Hence the selected entries of the input vector can be zeroed. SInce each of these Givens rotations are orthonormal matrices. Their cascade is an orthonormal

matrix as well. Additionally as can be seen in Equation 3.11 the observation vector b should also be rotated with each of the Givens rotations so that the c vector is also available once the QR decomposition is finalized.

The progress of a QR decomposition based on Givens rotations zeroing out one element of the matrix at a time is illustrated in Figure 3.2.

It has twice the number of operations as the Householder reflections, that is $4n^2m - 4/3n^2$ flops [22]. Furthermore, floating-point operations are needed to compute the angle of a complex number and rotations with this angle. However, a CORDIC unit [25] suffices to implement these rotations in fixed-point and significantly simplifies the hardware implementation.

$$
G(i, k, \theta) =
\begin{bmatrix}
1 & \dots & & 0 & \dots & 0 & \dots & 0 \\
\vdots & \ddots & & \vdots & & \vdots & & \vdots \\
0 & \dots & & c & \dots & s & \dots & 0 \\
\vdots & & & \vdots & \ddots & \vdots & & \vdots \\
0 & \dots & & -s & \dots & c & \dots & 0 \\
\vdots & & & \vdots & & \vdots & \ddots & \vdots \\
0 & \dots & & 0 & \dots & 0 & \dots & 1
\end{bmatrix}
\begin{matrix} \\ \\ i \\ \\ k \\ \\ \end{matrix}
\qquad (3.13)
$$

$$\quad\; i \qquad\quad k$$

where $c = \cos(\theta)$ and $s = \sin(\theta)$

As will be discussed more in Section 4.1, the Givens rotation based method is chosen for the QR decomposition. The fundamental reason for this selection is that Givens rotation based method is much simpler to realize in hardware. That is Givens rotations can be implemented by CORDIC units whereas Householder reflections require computations of square of vector norms and divisions with these squares.

3.3   Hop-TERRAIN Localization Algorithm

After the method of triangulation have been selected, a representative implementation is deemed useful.For such an implementation as stated in Chapter 1 the key inputs are the reference node positions and the distances to these nodes. However a ranging system is not available. Therefore a simpler substitute metric is desirable. At this point turning to the two phase localization algorithm proves beneficial. As was discussed earlier the first phase of this algorithm, for purposes of coverage and initialization, the number of hops to reach from a reference node (also called hop count or hop distance for short) is used instead of the real Euclidean distance between the two nodes. Implementing such a localization algorithm serves two purposes.

- The triangulation system can be designed, deployed and tested even with the absence of a real Euclidean distance measurement or ranging system. That is the ranging aspects of localization will be realized by counting the number of hops from reference nodes.

- When the range measurement system is available the hop distance metrics are simply replaced with real distances. Also in a final implementation the same triangulation unit can be utilized first during the initialization phase with the hop distance metrics as well as reference node positions as inputs. Next during the refinement phase, when ranges are available, inputs to the same unit can be immediate neighbor positions as reference points and the Euclidean distances to these neighbors. A detailed discussion and analysis of this algorithm along with its simulated performance can be found in [17, 19].

Since the hop count is used instead of real distances, hop count or hop distance determination is in effect the range measurement mechanism for the implemented ranging system. The initiation of these hop count measurements are conducted by

the anchor nodes , which are the reference nodes that have positions preprogrammed during installation.

3.4 HopTerrain Implementation Issues

To determine the hop counts at each node, the anchors periodically initiate broadcast messages that include the position of the anchor and a hop count equal to 0. Their immediate neighbors receiving this broadcast relay it to their neighbors with the hop count incremented by one. Hence, the messages initiated by the anchors propagate throughout the network with increasing hop counts. This process is also called flooding of the network.

Anchors periodically start new rounds of flooding to capture any additions to the network and track possible position changes. When a node in the middle of the network receives messages from 4 or more anchor nodes it knows the position and hop distance of 4 reference points. Therefore once it has enough information to perform a triangulation it computes its position. As the node hears from more number of anchors it retriangulates and updates its position.

Also periodic repetitions of this flooding procedure tries to ensure tracking the dynamic aspects of the network. This periodic repetitions are realized by anchor nodes periodically sending out flooding messages into the network. The period of these rounds of flooding are determined by a counter set at the anchor nodes. That is every time the counter expires the anchor sends out a new flooding message. The limit of this counter can be programmed via software making the intermission between rounds of flooding programmable. To distinguish between different rounds of flooding an ID is assigned to each round.

To realize the proposed HopTerrain localization algorithm a number of auxiliary

functions and structures need to be implemented. These include memories for storing anchor information, defining packets that contain localization information and can be transmitted within the radio protocol stack. Units that can encode and decode these packets, etc...

Flooding messages are communicated throughout the network using an additional type of data packet called localization packet. The structure of localization packets is illustrated in Figure 3.3. These packet consist of 7 bytes. First two bytes are used by the DLL and are not delivered to the localization block also during transmission they are appended by DLL. They are the node ID and packet length. The payload is 5 bytes long. first byte contains a 5bit hop count and a 2 bit packet type. the next 3 bytes are the x, y, z coordinates of the positions, each 8b long. Final byte is the flooding ID which is a byte long.

Upon their detection at the Data Link Layer these packets are passed on to the localization subsystem for decoding and any subsequent action. Also, the received flooding messages are relayed by incrementing the number of hops, creating a new packet and passing that to the Data Link Layer for transmission to immediate neighbors.

Additionally there is a parameter that specifies the node is an anchor. This parameter is also programmable via software. If the node is an anchor its coordinates as well as the period between flooding rounds need to be programmed during the startup of the node. Since anchors are the effective initiators of the Hop Terrain localization scheme their presence is critical for any test of the localization system.

There are two types of localization packets as mentioned in Figure 3.3 flooding packets and maintenance packets.

- The flooding type packets are those initiated at the anchors. Open their arrival, source anchor is searched in the list of known anchors. The subsequently taken

id	len	0 hc t	sl	fid

metadata   MAC payload

id:	**Node ID**	**8b**	
len:	**Packet length**	**8b**	
hc:	**Hop count**	**5b**	
t:	**Packet type**	**2b**	**0: fl,1:maint**
sl:	**Node location**	**24b**	**X,Y,Z**
fid:	**Flooding ID**	**8b**	
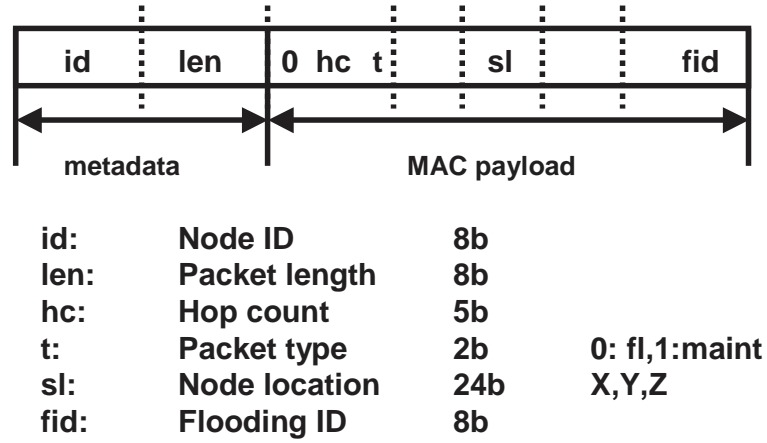
Figure 3.3: Structure of localization packets.

actions differ depending on the whether or not that anchor is present in the list of known anchors and parameters of that list entry.

– If the source is not in the list it is added to it and if total number of anchors is greater than four a triangulation computation is commenced also the flooding message is relayed to the neighbors.

– If the flooding source is in the list of known anchors then the relevant entry is modified only if the received hop count is smaller than the list entry or the flooding ID is larger than that in the list.If the list entry is modified also the flooding message is relayed and a triangulation in commenced.

– If the flooding source is known but the flooding ID is smaller to that in list entry, i.e. message is stale, or with same flooding ID if the hop count is larger the messages are ignored and not relayed.Also no triangulation is initiated.

• The maintenance type of messages are those initiated by immediate neighbors of the node. They are generated when a node finishes a triangulation and

obtains a, possibly new, position for itself. When such a message is received from a neighbor the relevant entry in a neighbor data storage is changed and the message is not relayed.

3.5 Conclusion

This section the design of a HopTerrain based localization system is presented. This algorithm is the first phase of a localization system and can be easily adapted to perform triangulations with Euclidean distances. The chapter started with a mathematical formulation of the localization system. Following this problem formulation methods on its solution are considered and QR decomposition based LSsolution is selected for realization. QR decomposition is realized with Givens rotations. Following the solution functional add on's that will allow computation of the HopDistances are discussed. Next chapter will discuss the implementation of the triangulation based HopTerrain localization system.

Chapter 4

System Implementation

The proposed localization system is realized in silicon as part of a sensor network protocol processor. The chip implements a protocol stack that is tailored for wireless sensor network applications. Subsystems generally follow the OSI reference model [26] and include the application, network, data link, and digital baseband portion of the physical layers [27]. The protocol stack is augmented to include the location subsystem proposed in the previous section so that the sensor node is capable of self localization.

A simplified block diagram of the implemented localization system is given below in Figure 4.1. The system has a Least squares equation solver, which has its own modules, a known anchor information storage list or anchor list for short, as well as receive (RX) and transmit (TX) sub blocks. Also note that the arrows in the Figure 4.1 illustrate data flow between these blocks. In the rest of this section these sub blocks are described in detail.
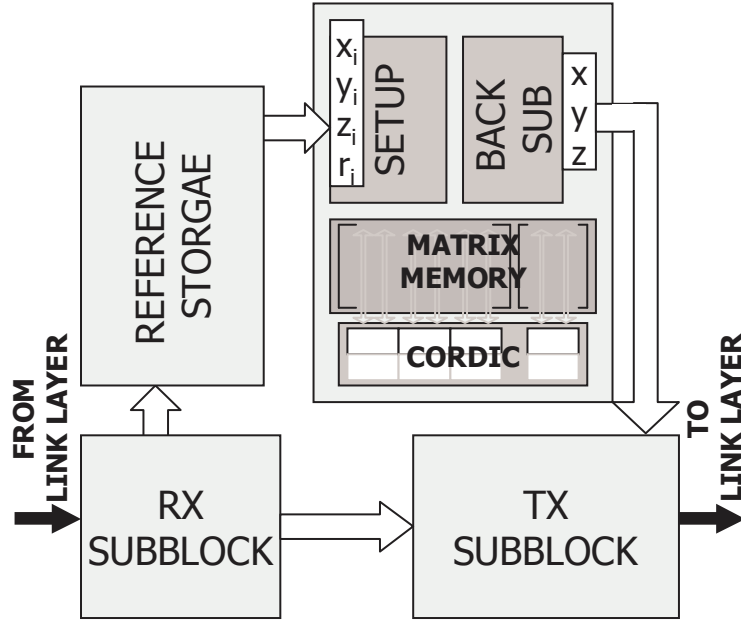
Figure 4.1: Simplified localization system block diagram.

4.1 Least-squares (LS) Solver

The Least squares solver is the most computationally intensive block in the localization subsystem. Therefore to optimize system performance its design and optimization received considerable effort. Besides achieving low power, there are computation rates that the Least Squares sub block must be able to support. The computation rate is predominantly determined by localization packet receptions. This depends on the rate that anchors start rounds of flooding. The flooding rate is programmable and can be as low as one flooding per tens of minutes. Assuming a conservative "one flooding round per minute", the rate of updates that needs to be supported would be at most 16 LS calculations in one minute. This corresponds to a time of 4 seconds per calculation. As will be seen later, this rate is low enough that all the computation is performed serially in time. In effect relaxed timing is traded for hardware complexity.
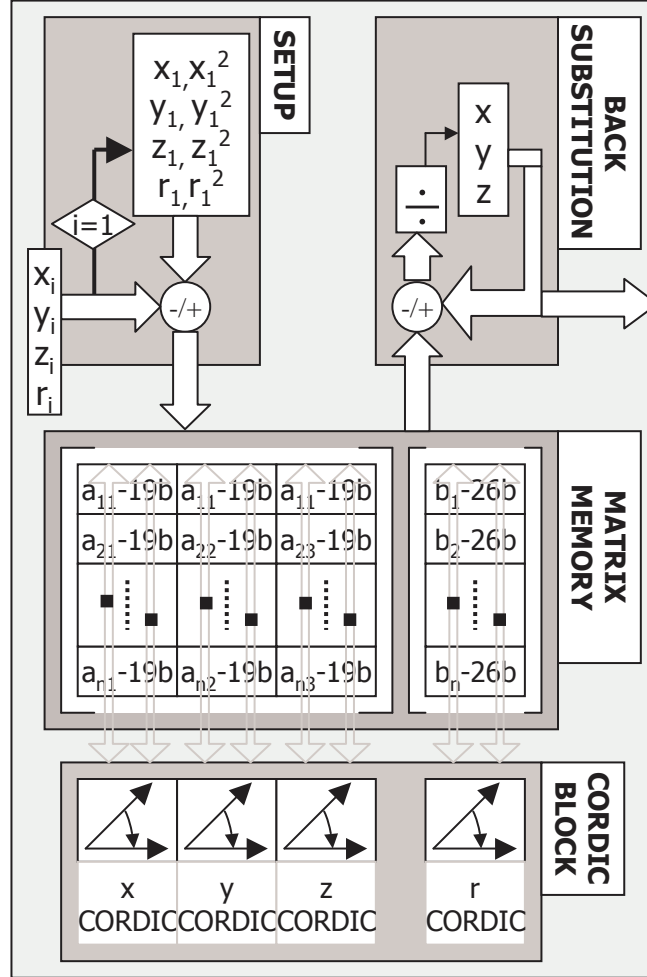
Figure 4.2: LS solver block diagram

One set of CORDIC blocks is used to decompose the matrix one element at a time and still meet the timing specification. Figure 4.2 shows the block diagram of the LS solver block. The subblocks of the Least squares solver are CORDIC unit, the setup block, matrix memory and back substitution unit.

4.1.1 CORDIC unit

As discussed in previous sections the triangulation block computes the LS solution via QR decomposition. In addition it preforms QR decomposition via Givens rotations. Givens rotations are best impemented in hardware using CORDIC units.

CORDIC is a recursive computation that is used to rotate vectors and has been around for 50 years [25]. The core idea can be explained by looking at simple rotation equations that define a vector $[x_{in}, y_{in}]$ being rotated by an angle of θ, the components of the resulting vector $[x_{out}, y_{out}]$ would be,

$$x_{out} = x_{in} \cos(\theta) - y_{in} \sin(\theta)$$
$$y_{out} = y_{out} \cos(\theta) + x_{in} \sin(\theta)$$

(4.1)

factoring out the $\cos(\theta)$ terms would yield

$$x_{out} = \cos(\theta)(x_{in} - \tan(\theta)y_{in})$$
$$y_{out} = \cos(\theta)(x_{in} \tan(\theta) + y_{in})$$

(4.2)

Approximating θ as sum of a series of angles where $\theta_i = \arctan(2^{-i})$ the associated multiplications become multiplications by 2^{-i} and can be implemented as right shifts.

$$x_{i+1} = K_i(x_i - 2^{-i} \times y_i)$$
$$y_{i+1} = K_i(x_i \times 2^{-i} + y_i)$$

(4.3)

$$K = \prod_{i=0}^{N-1} K_i = \prod_{i=0}^{N-1} \cos(\arctan(2^{-i}))$$

(4.4)

This multiplicative factor converges to ≈ 0.6 as number of stages increase. Hence given a vector and angle CORDIC can perform a rotation also called rotation mode operation, given a vector CORDIC can compute the magnitude and angle, which is

called vectoring mode. During operation this block is first used in vectoring mode to obtain the angle of the complex pair, next in rotation mode all four complex numbers are rotated by the angle obtained in the previous step.

Due to the non streaming nature of the localization data, the computations are bursty and infrequent. Therefore timesharing the minimum amount of hardware would prevent unnecessary leakage when the block is inactive. Hence the CORDIC unit is implemented in a time serial fashion as well. Figure 4.3 shows a CORDIC slice that can be used while rotating. When i = 0 the input vector and rotation amount is loaded into the registers and in subsequent iterations the sign of the angle is monitored such that it tends toward zero. However it should also be pointed that the rotated vector includes a factor given by Equation 4.4 which needs to be undone before the result is returned.

An N stage CORDIC computation is completed in N cycles. In this implementation a 10-step CORDIC is selected for implementation. Therefore one computation finishes in 10 cycles. The bitwidths and fixed point concerns of the CORDIC units are addressed in Section 4.6.

4.1.2 Other LS solver sub blocks

There are three other sub units within the LS solver block. These sub units are: Setup unit, Matrix memory and back substitution unit. Setup unit is functional for setting up the matrix to decompose. Essentially it computes the contents of the Equation 3.3 and Equation 3.5 from x, y, z and r values. The first reference point is registered and the values associated with this reference point are subtracted from the information of the rest of the matrices. In this block an 8-bit squarer from the synthesis library is utilized to compute the squared values. The result widths are extended such that overflow possibilities are prevented.
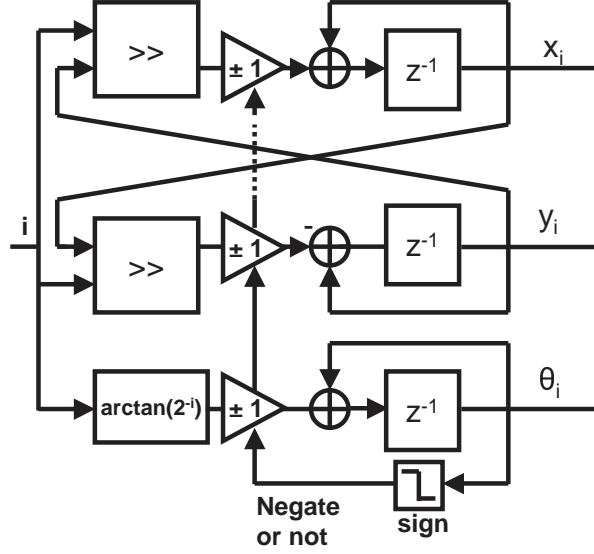
Figure 4.3: CORDIC block that iteratively rotates a vector by a given angle.

The matrix memory is a register file that can hold up to 16 words of data. The first three columns of its contents represent the matrix introduced in Equation 3.3 whereas the last column represent the vector in Equation 3.5. The bitwidths are shown in Figure 4.2 and 19b for the A matrix and 26b for the b vector, all designed to avoid overflow.

Once the matrix is in upper triangular form the result can be computed by a back substitution operation. that is for a 3×3 upper triangular matrix R composed of elements r_{ij} and a vector b with elements b_i and unknown positions u_x, u_y and u_z it can be written that

$$u_z = \frac{b_3}{r_{33}} \tag{4.5}$$

$$u_y = \frac{b_2 - r_{23}u_z}{r_{22}} \tag{4.6}$$

$$u_x = \frac{b_1 - r_{12}u_y - r_{13}u_z}{r_{11}} \tag{4.7}$$

Back substitution uses a fixed-point serial divider [28]. This divider converts the fixed point numbers to a mantissa and an exponent sections,just like in a floating point number, followed by a division of the mantissas. After the division the result mantissa is shifted by the result exponent to obtain the fixed point quotient.

The division module has a detection module for the case of a *"divide by zero"*. In this case the output is saturated to the maximum output and a *"divide by zero"* flag is asserted. Also for the case where the quotient exceeds the 8bit representation range it is saturated. the practical implication of such a situation is that, when the one of the coordinates turn out to be outside the intended grid, it is assumed to be on the edge of the grid.

4.1.3   Least Squares Complexity

The matrix to be decomposed is calculated using the data in the anchor list. That is, the matrix A of Equation 3.3 and vector b of Equation 3.5 are computed using the information anchor information from the anchor list. They are then stored on a matrix memory. Once the matrix to be decomposed is computed in the setup block the application of Givens Rotations commences. The CORDIC rotators are used first to determine the angle of rotation and then to apply the Givens rotations to the matrix. The CORDIC block performs each rotation in 10 iterations. It is also implemented time sequentially and therefore, each CORDIC rotation takes 10 cycles. After 20 cycles, nullification of one element is completed. At each iteration, the Givens rotation that nullify entries of the matrix A are also applied to the vector b on the right hand side of the equality. Hence the equality is preserved at all times.

In the worst case that all 16 anchors are used in the computation. Then there are 39 (14+13+12) entries that need to be nullified. Hence, 780 cycles are needed to complete the QR decomposition. Once an upper triangular matrix R is obtained, the

resulting 3-by-3 system is solved using back substitution. Since the back substitution uses a fixed-point serial divider and the quotients are 10 bit wide. A divide operation takes 10 cycles and three divides take a total of 30 cycles. Another 15 cycles are also spent at the set up stage as the matrix memory is loaded. Therefore, in the longest case, approximately 825 ($= 780 + 45$) cycles are used during the LS computation step.

The Digital Sensor Node IC is designed for 16MHz operation. At this rate, the 825 cycles take only about 52μs. Including the matrix multi-cycle divisions and other multi-cycle operations, the total time goes up to 70 μs. This computation time can easily satisfy the timing requirement discussed earlier. The design of the localization system is performed in VHDL. Thereafter the design is synthesized, placed, routed and extracted. Following the parasitic extraction and back annotation, post layout simulations estimate 1.7mW average active power dissipation and 0.122nJ of energy consumption during the simulation. These simulations use the 16 MHz clock rate and a 1.1V supply voltage in active mode. An equally important metric is the energy consumption per flop in the localization system. Using the energy number and floating point operation count of 1000, which is calculated below, the energy per flop estimate is 0.122nJ/op. It was stated in the previous section that QR decomposition with Givens rotations requires $4n^2m - 4/3n^3$ flops. With m $= 15$ and n $= 4$, the expression above yields 880 flops. Including the setup of the matrix A and vector b of Equation 3.5 as well as the final back substitutions, the operation count goes up to 1000 flops. It should be noted however that the power and energy numbers quoted above are optimistic estimates since they do not consider the dissipation at any of the RX and TX sub blocks. However since the main contributors, the LS solver and the anchor list, are included in the simulations, the discrepancies would be negligible.
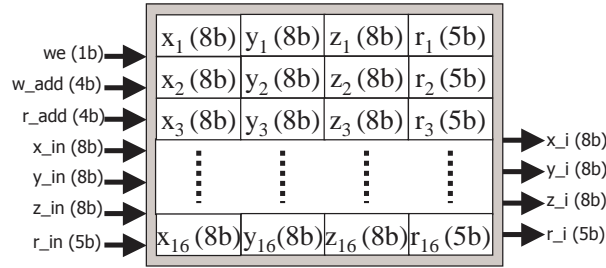
Figure 4.4: Anchor List data storage

4.2   Anchor List

The second sub block of the localization system is dedicated to maintaining the information regarding anchors known to the node. This subsystem includes a data storage that highly resembles an array of anchor objects (Object in the object oriented programming sense), where each anchor object includes the anchor coordinates and its hop distance. Figure 4.4 illustrates a block diagram of the implemented anchor list.

For the particular implementation considered in this work, the maximum number of anchors is chosen to be 16. This number is selected because in a network of possibly hundreds of nodes it requires a manageable effort in preprogramming the anchor nodes. As a power of two it also allows efficient data addressing.

4.3   RX and TX Sub-blocks

In addition to the sub-block that maintains the anchor list, there are two more sub-blocks worth mentioning. First of them handles the reception and decoding of localization packets from the Data Link Layer. The second sub-block handles the creation of the localization packets and their delivery to the Data Link Layer. The
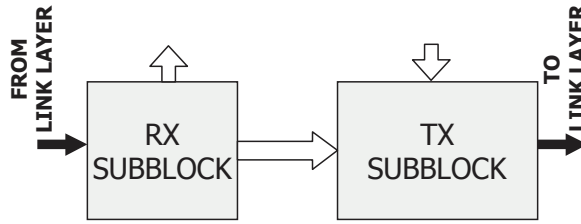
Figure 4.5: RX and TX subblocks

position of these sub blocks is depicted in Figure 4.1. The TX sub-block also handles the small tasks such as incrementing the hop count before relaying a flooding message or setting it to zero if the node is an anchor.

4.4 Design Methodology

Unlike conventional digital implementations this design is not realized straight up in an HDL. The design of datapath components are initially done in Matlab Simulink® fixed-point blockset. Additionally Matlab Stateflow® is used to design the controller finite state machines. With this setup initial overflow, timing and fixed point performance degradation effects are observed as well.

Once correct functionality is achieved in Matlab, the datapath components of the design is hand translated into a modeling language called *"Module Compiler Language (MCL)"*. This language is the input for a Synopsys® tool called Module Compiler® .This tool can produce synthesizable RTL for many different kinds of datapath blocks,such as different kinds of adders, multipliers, dividers etc. The MCL language is a higher level language than VHDL or Verilog and it is supposed to simplify architectural explorations of datapath modules. Designs in MCL can either directly be consumed by the logic synthesizer for physical design or can be translated

in to behavioral or gate level HDL for simulation.

After the first functional verification in Matlab, the state machines that were designed in Stateflow are directly converted into VHDL using an inhouse developed tool called SF2VHD [29]. Once the block datapaths are available in MCL and state machines are available in VHDL these two units are instantiated and connected inside a VHDL entity. Subsequent simulations in an HDL simulator, such as Modelsim® aims to verify the design by comparing it to the reference design in Matlab Simulink. The rest of the block integration as well as glue logic additions are all performed in VHDL and is verified again on an HDL simulator.

However for multi node verifications an emulation environment is the preferred method. Simulating multiple nodes in an HDL simulator which run its own software is a computationally intensive and resource draining process. Therefore an FPGA based emulation environment was utilized for faster verification [30]. Once the VHDL description of the protocol processor is available it is simply synthesized with an FPGA library so that the protocol processor description can be loaded into an FPGA. An inhouse Emulation Environment called the Berkeley Emulation Engine [31], which consists of upto 16 FPGA chips inter connected with reconfigurable connections, are used to emulate a network of sensors. Any selected signals from the protocol processor and these signals from the FPGA are monitored by use of a logic analyzer.

This emulation setup is initally developed to verify the Data Link layer functionality of the Protocol Processor [27, 30]. Later on it is modified to verify the localization block functionality and achieved remarkable speedups compared to multi node simulations in an HDL simulator.

4.5 Alternative implementations

4.5.1 First Alternative Implementation: General purpose Microprocessor (μP)

One can argue that with such relaxed timing requirements the localization system can be realized using a general-purpose microprocessor (μP), which would ideally be embedded on the Digital Sensor Node IC. The general-purpose computation alternatives were dismissed early on due to their expected power consumption. However this decision still needs to be quantified. The power dissipation of general purpose μP are usually published characteristics of these components. The energy consumption however depends on the length of operation. Hence, the duration to compute LS solution on such devices should be computed. The μP implementation of the localization is better equipped to compute the LS solution via Householder reflections. It is known from previous sections that 40 flops are needed for the QR decomposition. However additional clock cycles are needed for retrieving the anchor data from the memory into the data cache, setting up the matrix and back substituting the triangular matrix. Also multi-cycle division and square root operations also add more cycles to the operation. However many embedded μP do not have floating point hardware support [32]. These chips would execute fixed-point software to emulate these floating-point operations. As an example, the embedded μP in [32] dissipates 450mW at 600MHz. In this processor the LS computation takes 820 clock cycles or 1.368μs to finish the computation. Therefore the energy consumption becomes 547nJ for 560 flops. This yields a 0.977nJ/flop Next consider a 1MHz, 16-b microcontroller (μC) that is targeted for low power applications [33]. Reported active power dissipation is 0.9mW. Some multi cycle computations should be expected due to 16b hardware used to perform computations on the 19 and 26 bit data shown in Figure 4.2. Including all

	Time(μs)	Power	Energy	Energy/flop
This work	71μs	1.7mW	122nJ	0.12nJ/flop
Embed'd μP[32]	1.37μs	450mW	547nJ	0.98nJ/flop
Embed'd μC [33]	2.5ms	0.9mW	2.25μJ	2.25nJ/flop
μP w/ FP accel. [34, 35]	2.7μs	952mW	2.57μJ	4.59nJ/flop
DSP w/ fp support [36]	4.12μs	825mW	3.41μJ	6.01nJ/flop

Table 4.1: Estimation of energy/flop metric for various implementations.

expected overheads QR decomposition with Givens rotations requires 2500 cycles (or 2.5ms) to complete. Over this period the consumed energy is 2.25μJ and the effective energy/flop metric yields 2.25nJ/flop For a low power system with floating point support, consider the chip set proposed in [34] and [35]. It consists of a processor and a floating-point accelerator, where the first dissipates 800mW [35] and later dissipates 152mW [34]. The clock rate is 250MHz. The computation time is calculated to take up 686 clock cycles or 2.7μs. The total energy consumption is 2.57μJ for 560 flops and the energy perx operation is 4.59nJ/flop.

Finally a commercially available digital signal processor (DSP) with on chip floating point addition and multiplication support [36] is considered. The chip is clocked at 150MHz hence completes the computations in 620 cycles or 4.13μs. The reported power dissipation is 825mW [37] and the energy dissipation is computed 3.41μJ for 560 flops. Therefore for a DSP implementation energy/flop metric becomes 6.01nJ/flop. As can be seen in Table 4.5.1 these alternative implementations consume an order of magnitude higher energy than our dedicated hardware implementation.

4.5.2 Second alternative implementation: Dedicated, parallel, pipelined, systolic implementation

Dedicated digital signal processor systems and circuits solving QR factorization problems have previously been reported in the literature. They are used for the adaptive nulling in multiple antenna receivers [38] and adaptive signal processing in implementing the square-root Recursive LS (or RLS) algorithms using QR factorization [39]. In both cases the final solutions involved fully parallel and pipelined solutions known as systolic arrays. However it should be noted that the inputs to both systems are continuous stream of input samples. The flow of samples can fill up a computation pipeline and allow pipelining the data flow. Therefore, the use of systolic arrays, which implement a pipelined form of parallel computation, is an appropriate choice. In contrast, distributed Sensor network localization problem requires infrequent LS computations. A pipeline would never be completely full and the operation would be inefficient. Hence a parallel implementation would be inappropriate for this case.

4.6 Performance Penalty Due To Fixed-Point Implementation

The design of the localization system was performed in VHDL using fixed-point arithmetic. The position information is represented as 8-bit signed numbers for each coordinate. The hop count is an unsigned 5-bit number. With 8 bit coordinates, an indoor network locality with 100m edges would have 0.5m of resolution. Considering that one hop can correspond to up to 10m of real distance [15], 0.5m of coordinate resolution error would be smaller than error caused by hop counts used as crude distances. The degradation from a floating point least squares solution to a fixed

point least squares solution is 6The floating point LS solutions were computed with
MATLAB while the fixed-point results were obtained by cycle-true and bit-true HDL
simulations. To avoid high error due to fixed-point implementation, when necessary,
the word lengths (or bit widths) were increased and outputs were scaled down. In the
remaining of this section, this will be explained in more detail. Contents of the matrix
to be decomposed (A in Equation 3.3) are sums or differences of the 8-bit position
inputs. Therefore they need to be represented by 9 bits. Similarly, the elements of
the result vector (b of Equation 3.5) require 19 bits for each entry. This can be seen
as follows. Squared 8-bit numbers become 16-bits numbers and sum of six 16-bit
numbers require 19 bits to avoid overflow. There is also data expansion possibility
due to CORDIC units. After one rotation an output can increase to v2 of its value.
Such a case occurs when a vector with both coordinates equal to the highest k-bit
number (2k-1-1) is rotated such that one of its coordinates becomes zero. In this case
the other coordinate becomes equal to the magnitude of the vector. Since the matrix
being decomposed has 3 columns, any of its entries can be rotated at most 6 times.
Hence the value of the Matrix entries can increase by v26 = 8 times. This corresponds
to a 3 bit wider output than the input. In addition, the 10-stage CORDIC involves
10 additions in total. These additions can require ceil(log210)=4 more bits. At the
end of each rotation the CORDIC results are normalized so that this need for 4 bits
would remain internal to the CORDIC rotation and not accumulate to any subsequent
rotations. In summary, 3 bits are needed for result overflow avoidance and 4 bits are
needed for internal CORDIC rotation overflow avoidance. Therefore a 7-bit increase
is needed during the QR decomposition stage. This requires the matrix entries be
sign extended from 9 to 16 and the vector entries to be extended from 19 to 26
bits. These are also the word lengths used in the fixed point implementation. In the
back substitution, divisions are performed to yield 10-bit quotients and these 10 bit
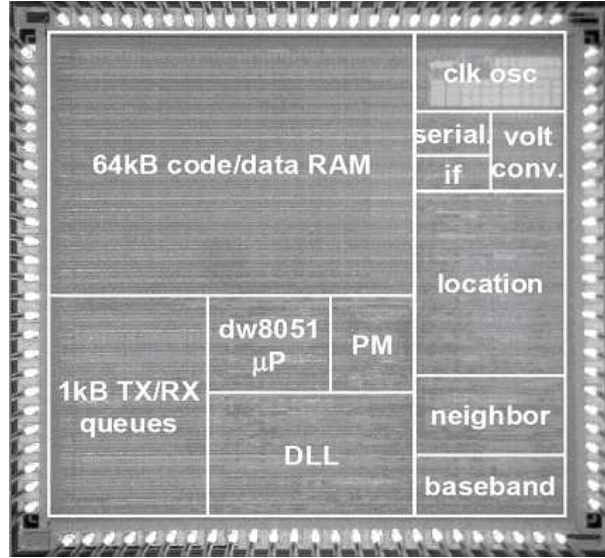
Figure 4.6: Micrograph of the Digital Sensor Node IC. Localization system is on the middle right.

outputs are saturated down to 8 bits. The mean position error increases significantly in cases that the 10-bit quotient needs to be saturated during conversion to an 8-bit number. During the simulations these case were encountered les than 5Physically, a result that needs to be saturated implies a node falling outside the grid intended for the network and this node is assumed to be on one edge of the network. Such cases can be prevented with careful anchor topology planning. Especially positioning the anchor nodes uniformly on the periphery of the network is such a solution proves to be helpful also in other aspects of localization problems [19]. Such an approach would remedy calculated location errors due to saturated results.

4.7 Physical Implementation

The Digital Sensor Node IC, one of whose sub blocks is the localization system, has been implemented on silicon using a 0.13μ CMOS manufacturing process. The

chip has dimensions of 3mm by 3mm. The Localization system is located on the lower left corner of the Digital Sensor Node IC and occupies an area of 0.79mm^2. The final layout is shown in Figure 4.6 and the localization system physical implementation results are summarized in Table 4.2. Note that power estimates include all parasitic effects.

The leakage currents numbers are done in three cases:

Gating disabled: 30.2μA @1V

VRC Gating enabled (vddlo=1V-V$_{tnmos}$): 19.1μA @1V

DRV Gating enabled (vddlo=300mV): 954nA @ 0.3V

There's a large power savings because it's much more effective to gate logic than memories, and your block doesn't have any memories in it. Finally it should be pointed out that even better leakage reductions can be achieved when the block is powered off in standby mode.

4.8 Conclusion

An integrated and low power localization system implementation has been presented. The system serves as part of a distributed self-configuring, ad-hoc sensor network node. It calculates the sensor node position via triangulation. Triangulation is performed by computing a Least Squares solution. Various alternative implementations were considered. A QR decomposition using Givens Rotations was selected for the LS solution algorithm. The final design exhibits 1.7mW of active power dissipation. This implies an order of magnitude active power dissipation savings with respect to a General-purpose microprocessor or DSP implementation. This shows the low power dissipation goal of the implementation has been achieved. In addition the

Parameter	Value
Manuf Process	0.13μ CMOS
Area	0.79mm^2
Dimension	$820\mu \times 970\mu$
Gate Count	30k
Register Count	3k
SIm Power(including parasitics)	1.7mW
Clock Freq	16MHz

Table 4.2: Localization System Physical Implementation results

system occupies 0.79mm^2 of silicon area and the fixed-point implementation causes a negligible degradation in the accuracy of the final location outputs.

Chapter 5

Ranging in Sensor Networks

The second fundamental task required to implement the selected localization algorithm is ranging or measuring the distances between sensor nodes. The localization system would be complete once it has the ranging system in addition to the triangulation unit. One key difference between these two units is that designing a ranging unit is a much more open ended question than designing a triangulation unit. For instance realizing the triangulation unit is essentially a least squares solver implementation. On the other hand the ranging unit can realize even different algorithms let alone digital implementations. Hence the design of the ranging unit needs a more detailed top down consideration, starting from the fundamentals of the problem and slowly refining to a hardware implementation. Along this line the second part of this thesis will begin by considering the fundamental relations between wireless channel and ranging. In subsequent chapters a ranging system will be proposed to realize the selected method and this system will be implemented and prototyped.

In this current chapter different channel views will be compared and contrasted against each other regarding their feasibilities for ranging. Accuracy,robustness and implementation considerations are important criteria for these comparisons. Finally

the preferred method will be revisited to address its shortcomings.

5.1 Wireless channel views used for ranging

Ranging can be seen as exploiting the properties of the wireless channel to measure the transmitter(TX) receiver(RX) distance. This task can be achieved by looking at the channel from different views to estimate the ranges. Even though it is the same wireless channel, seen through different views of the channel, its different properties are utilized or emphasized. Therefore performance, robustness and implementation issues of these methods vary significantly from each other. Existing ranging methods fall into three main categories that differ in the way the channel is viewed:

- Time domain view

- Frequency domain amplitude view

- Frequency domain phase view

The reason for the possibility of using these different methods for ranging is that the transmitter to receiver distance involved in the wireless channel can manifest itself in different properties of the channel. Each of these views deserve a discussion on their own and will be considered in detail.

5.1.1 Time Domain View

In the time domain view, the channel is described by its channel impulse response (CIR). In the absence of multipath arrivals [40], the CIR is a scaled and delayed impulse.

$$h(t) = C_0 \times \delta(t - \tau_0) \tag{5.1}$$

where τ_0 is the line of sight (LoS) Time of Flight (ToF) and C_0 is the signal amplitude. Without multipaths, the channel can also be said to have a single tap. In the presence of multipaths, transmitted signal arrives at the receiver via multiple propagation paths at different delays [41, 42]. Therefore, additional channel taps appear in the CIR along with the LoS tap.

$$h(t) = \sum_{i=1}^{N} C_i \times \delta(t - \tau_i) \tag{5.2}$$

where, N is the total number of multiple paths, C_i and τ_i are the amplitude and the ToF of the signal arriving through i^{th} propagation path.

It should be noted that, even in multipath conditions the LoS path would remain as the term with the smallest τ_i in Equation 5.2. Therefore, estimating the CIR would allow determining the LoS ToF. It can be argued that dielectric media, e.g. glass on the LoS path can change the ToF due to the different light speed within it. However in practice these media on LoS path with different light speed would not constitute a significant portion of the path length. Therefore the LoS arrival can be assumed all over the air.

To realize such a ranging system effectively a CIR estimator needs to be implemented. Channel estimation is a common need arising in many different communication systems and a rich literature exists on this topic [41, 42, 43].

CIR estimation, or commonly known as channel estimation, is usually studied in three categories:

- Non Data Aided (NDA)

- Decision Directed (DD)

- Data Aided (DA)

Non Data Aided(NDA) channel estimation does not require any pilot or known data. Instead it uses some assumptions about the statistics of the input data such as its whiteness. It also employs sophisticated signal processing techniques like LMS-Kalman Adaptive algorithms [43] and Maximum Likelihood (ML) Detection [44]. Decision Directed(DD) methods use decoded data symbols to assist the channel estimation problem and reduce its computational complexity. Both NDA and DD channel estimation methods require certain SNR levels to start and maintain functionality. Data aided(DA) channel estimation methods use the presence of known data symbols and extract the channel parameters by using these terms and sequentially use these parameters for decoding. If perfect training sequences, without any random data, are used then the channel estimation becomes equivalent to matched filtering with the given pilot data [43]. By matched filtering, the received signal with the pilot sequence the communication channel can be estimated.

Many ranging systems use this channel view including GPS, ultrasonic transducer based and UWB-based systems [45, 21, 46, 47]. Several factors affect the accuracy of this method. First, the resolution is proportional to the bandwidth [44]. Thus, the signal bandwidth and sampling rate directly affect the achievable accuracy. Additionally, the LoS tap may have a time shift if the transmitter (TX) and receiver (RX) clocks are not synchronized.

5.1.2 Frequency Domain Amplitude View

The frequency domain amplitude view considers the power attenuation due to propagation. In the absence of multipath, the received signal strength (RSS) behaves as

$$|H(\omega)| \ \propto \ 1/d^{\beta} \tag{5.3}$$
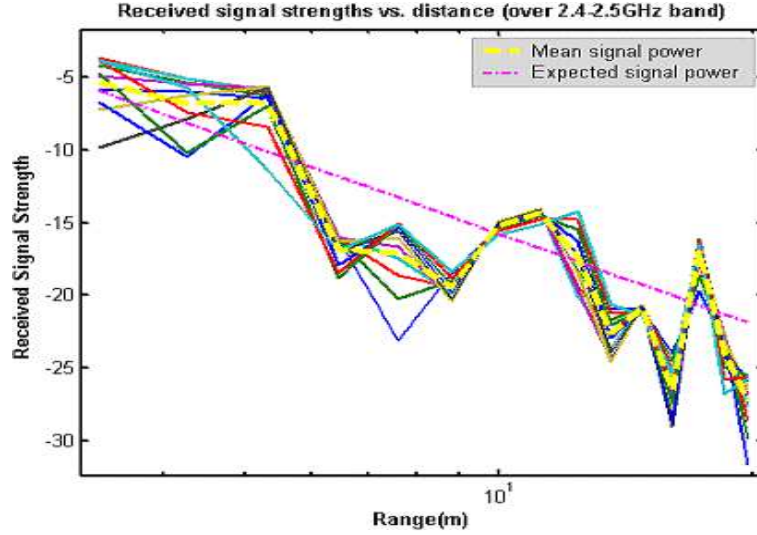
Figure 5.1: Effect of indoor wireless channels on the RSS over different distances

where, d is the range and β is the propagation factor [40]. Therefore, ideally, if the received power of a signal with certain transmission power is measured the range can be estimated per Equation 5.3.

However in presence of multipath, signal components arriving via different paths may interfere destructively [41] or constructively. Then, the channel behaves like a filter and the channel frequency response exhibits additional attenuation and amplification in its spectrum. Thus, as can be seen in Figure 5.1 the overall signal attenuation is not solely due to propagation anymore but also to the channel frequency response.

Due to the dependency of the RSS on the channel frequency response, these measurements require extensive calibration steps for functionality [48, 10].

Implementing ranging systems using this channel view require measuring the received signal power. Therefore, this method has very low hardware overhead for a sensor node, as the raw RSS measurements are often already available from the data communication radio [49].
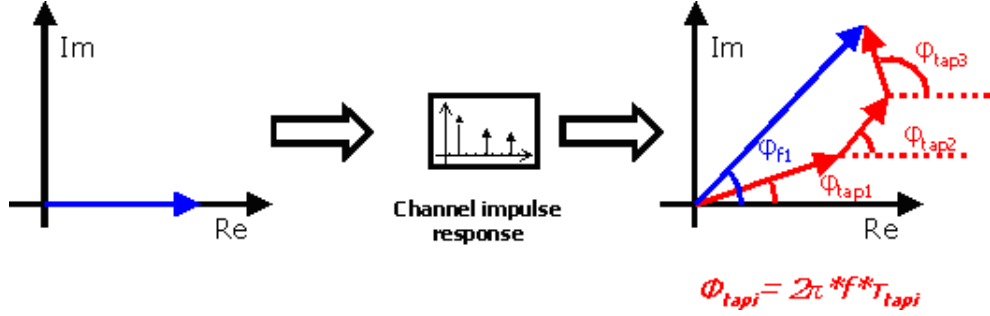
Figure 5.2: Illustration of an input phasor (left) transmitted through the wireless channel. Output phasor (left) is obtained as the vector sum of phase shifted and scaled copies of the input phasor

5.1.3 Frequency Domain Phase View

The frequency domain phase view utilizes the phase shift in the frequency response for ranging. Without any multipath, if the frequency domain representation of Equation 5.1 is considered, it is straightforward to observe that the phase response has a linear dependence on ToF

$$H(\omega) = C_0 \, e^{j\omega\tau_0} \tag{5.4}$$

$$\angle H(\omega) = \omega\tau_0 \tag{5.5}$$

where ω is the frequency and τ_0 is again the LoS ToF.

With presence of multipath the simple linear output phase dependency to the ToF disappears. In this case the output phase is again determined by the impulse response of the wireless channel. Suppose the input signal is a cosine signal. Then, it can also be represented as a phasor with zero phase. As this phasor is filtered through the channel, each multipath component generates a scaled and delayed version of the input signal. Hence the output phasor is the vector sum of the vectors defined by the multipath arrivals. This effect is also illustrated in Figure 5.2.

The evident result is that the phase does not depend on the ToF of the shortest multipath arrival but also the impulse response of the wireless channel. Therefore to realize a ranging system that utilizes the phase view of the channel, a phase measurement system as well as a channel estimator needs to be implemented. On the other hand once a channel estimator is in place it can be used to obtain the CIR and the time domain view of the channel can be implemented.

However *if the multipath effects can be neglected* this channel view has a simple implementation using quadrature phase demodulation. This condition is sometimes satisfied for acoustic or ultrasonic systems. Therefore there are instances of such systems using the phase view of the wireless channels [50, 51]. Recently some sensor network research also attempted utilizing phase information for ranging purposes as well [52]. Nevertheless, the presence of multipath in many RF signal based systems proves the phase domain view of the channel difficult to employ.

5.2   Selected Method and its Issues

In this work, time domain view of the wireless channel is chosen to implement the ranging system. The primary reason is the resiliency of this method against multipath effects. Since most sensor node applications require indoor deployment, multipath effects are expected to be common [40] and the ranging method needs to be robust when such effects are present. The rest of the section addresses issues with the accuracy, bandwidth, signal type and synchronization for the time domain view. These issues were identified as the key drawbacks of the time domain view of the channel and therefore deserve proper attention.

5.2.1 Limits on the ToF Estimate Accuracy

The Cramer Rao Lower Bound(CRLB) is an information theoretic bound on parameters estimated in presence of noise [44]. It defines a lower bound on the variance of estimation error below which the estimated quantity can not achieve. This bound for the ToF estimate variance (Var(ToF)) has been derived for UWB radar systems and is known to be:

$$Var(ToF) = \frac{1}{SNR}\left(1 + \frac{1}{SNR}\right)\frac{1}{\omega^2} \tag{5.6}$$

where, SNR is the signal to noise ratio and ω is the signal bandwidth [44]. From this equation, it is clear that increasing the bandwidth improves the ToF estimate.

CRLB defines the absolute minimum error variance that can be obtained with a certain estimator. However there can be additional factors that limit the estimator performance. The most important example of this issue arises when a sampled system is considered. In this case, the ToF is quantized to multiples of the sampling period (T_s). Then time quantization can also limit the ToF estimate accuracy. Assuming the ToF quantization error is within $e_{ToF} = \pm(T_s)/2$, the corresponding range error due to sampling becomes

$$e_{range} = e_{ToF} \times c = \pm\frac{1}{2F_s \times 10^6}300 \times 10^6 = \pm\frac{150}{F_s}[m/Msps] \tag{5.7}$$

where, $F_s = 1/T_s$ is the sampling rate in Msps and c is the speed of light. For a signal sampled at F_s the widest signal bandwidth that can be accommodated is $F_s/2$. Assuming a 10dB SNR for such a signal, the CRLB can be computed as

$$CRLB(SNR = 10dB) = \frac{1}{10}\left(1 + \frac{1}{10}\right)\frac{4}{[2\pi F_s]^2} \approx \left(\frac{0.105}{F_s}\right)^2 \tag{5.8}$$

$$\sigma_{ToF} \approx \frac{0.105}{F_s} = 0.105T_s \tag{5.9}$$

$$\sigma_{range} = \sigma_{ToF} \times c \approx \frac{0.105}{F_s \times 10^6}[300 \times 10^6] = \frac{30.45}{F_s}[m/Msps] \tag{5.10}$$

where σ_{ToF} is the deviation of the ToF measurements, σ_{range} is the deviation of range measurements and c is the speed of light in vacuum. Hence it is seen that even though CRLB is a lower bound on the ToF estimate error, the timing error due to the sampling effects has a larger variance than the CRLB and is a more restrictive bound on the ToF estimation performance. Incidentally this is also true for the low SNR case as obtained in Appendix A.

Given this result the ranging error due to the sampling effect is more restrictive and is used as the main design relation. Using Equation 5.7 and the ranging error specification of ± 1 m from Chapter 1, a bandwidth need of 75MHz is computed. The minimum sampling rate is thus 150Msps. This shows that sensor network localization can be realized with less than 100MHz bandwidth and without UWB (> 500MHz) signals. This is an important result for two reasons.

- The signal can fit in the 2.4Ghz ISM band.

- The signal is not necessarily subject to the Spectral power mask requirements of UWB specifications from FCC [53].

5.2.2 Wideband Signals

The next issue that needs to be addressed to implement a ranging system utilizing the time view of the channel is determining a wideband signal that can yield the desired accuracy.

Several types of wideband signals can be utilized in a ToF ranging system. Alternatives include:

- Pulse based Wide Band signals

- Pseudo Random Noise signals

- Chirp Signals

- Multi Carrier Wide Band signals

Using each of these signals for the ToF measurements necessitate the use of different signal processing algorithms with different complexities.

Pulse based Wide Band signals

The first implementation alternative uses pulse-based signaling. This method uses wideband gaussian shaped pulses and performs a frequency estimation on data already converted to frequency domain. As described in [46] the processing starts with the CIR being consisted of a sum of delta functions as in Equation 5.2. Taking the Fourier transform of this expression, the channel frequency response can be written as

$$H(\omega) = \sum_{i=1}^{N} C_i \; e^{(j\omega\tau_i)} \tag{5.11}$$

where, C_i and τ_i are the same as those in Equation 5.2. By formulating the channel in Frequency domain the channel estimation problem is converted into a classical spectral estimation problem, that is estimating the complex frequencies and weighing coefficients of superimposed coefficients [46].

Implementation of this algorithm requires a fast Fourier Transform (FFT) for converting the initial data sequence to frequency domain, Singular Value Decomposition (SVD) for separating the signal and noise. Also a Least Squares (LS) optimization algorithm is needed for estimating the frequencies. SVD alone is a computationally expensive operation with a cubic complexity i.e $O(N^3)$ [22] and should thus be avoided whenever possible. LS has an $O(N^2)$ complexity and the FFT has an $O(Nlog(N))$ complexity. Therefore, using pulse based signals with such an algorithm for ranging turns out to be computationally quite expensive.

PRN signals and Chirp signals

PRN and chirp-based signals are often used in Global Positioning Systems (GPS) and radars. When these kinds of signals are employed they can be matched filtered to estimate the CIR as was discussed in Section 5.1.1. Their correlation peaks are searched for time of arrival detection. Correlation of a received signal with periodic transmitted signal with period N is defined as

$$\phi_{rx}[n] = \sum_{i=0}^{N-1} r[i] \ x[i+n] \tag{5.12}$$

PRN signals are used in Code Division Multiple Access (CDMA) systems and GPS. They are defined to have near delta autocorrelation, where autocorrelation function is defined in Equation 5.13

$$\phi_x[n] = \sum_{i=0}^{N-1} x[i] \ x[i+n] \tag{5.13}$$

$$\phi_x[n] = \begin{cases} N & \text{if } n = 0 \\ -1 & \text{if } n \neq 0 \end{cases} \tag{5.14}$$

Therefore PRN signals have an autocorrelation that is easy to detect and are often used

Chirp signals are signals with time varying frequencies. They are used in radar systems to measure ranges. A basic chirp signal is a sinusoid with a linearly increasing frequency.

$$c(t) = cos((\dot{\omega}t + \omega_0)t) = cos(\dot{\omega}t^2 + \omega_0 t) \tag{5.15}$$

when either PRN or chirp signals are employed for ranging purposes the received signal is correlated(or matched filtered) with shifted version of its signature waveform. The time shift for which the correlation peaks is assigned as the time of flight. The complexity of correlation is less complex $O(N^2)$ [22] than SVD.

Multi Carrier based signals

Multi carrier based signaling operates by dividing the available spectrum into many narrow band channels or sub channels. Each sub channel can be treated as a separate narrowband channel. It can also be thought as a frequency division multiplexing method where many communication channels occur concurrently.

An alternative implementation of multi carrier signaling accepts the parallel data inputs (also called an OFDM symbol) in frequency domain and converts them into time domain using an inverse FFT (IFFT) operation [41, 54]. This technique is also called Orthogonal Frequency Division Multiplexing (OFDM).Then at the receiver by converting the received signal into frequency domain by using an FFT operation, the detection and demodulation is performed in frequency domain. Using an IFFT to generate the time domain signal obviates the use of multiple mixers. Additionally when communicating random data Inter symbol interference (ISI) can be avoided by placing guard intervals between OFDM symbols.

OFDM signalling is used in many recent successful communication standards such as IEEE 802.11a/g standards. In addition there are products available for Multi Band OFDM based Ultra Wide Band communication systems [55]. What is more there are upcoming wireless standards such as WiMax that further utilize OFDM signals within IEEE 802.16 wireless broadband access standard.

Such a signalling scheme also has attractions for ranging purposes. Instead of performing matched filtering in time domain using correlation based processing filtering can be executed in frequency domain. The main appeal of such an approach is the availability of the received signal frequency response at the receiver. If the transmitted data is known at the receiver the channel frequency response, can be obtained by simply dividing the received signal frequency spectrum with the transmitted signal frequency spectrum.

Once the channel frequency response is obtained an IFFT gives the CIR, which is the familiar time domain view of the channel. The required functions are an FFT to transform received signal to frequency domain, a division, and the final IFFT. Therefore the complexity of the processing is reduced to the complexity of the FFT algorithm ($O(Nlog(N))$) [22]. This is the least complex computation among previous processing alternatives which included SVD, LS and correlations.

Since there is no need to transmit random data in such a ranging system a periodic pilot OFDM symbol can be transmitted. This pilot symbol can be selected to match arbitrary frequency allocations and restrictions. Hence such a localization system can be reconfigured to operate in different frequency bands, accuracy requirements and with different spectral masks. Only the pilot sequence needs to be modified for this purpose.

In this work a Multi Carrier based signalling is selected using IFFT's for the signal generation. The main reason for this choice was the low computational complexity of this approach. Additional benefit of this scheme was its flexibility to generate signals that can operate over different bands and requirements.

Finally even though pulse processing based methods [46] and correlation based methods [1, 44, 47, 56] have received significant attention from the research community a multi carrier signalling based ranging scheme has received much limited attention previously. Nevertheless MB OFDM [55] wireless standard proposal suggests use of these signals for ranging purposes as well.

5.2.3   Synchronization

To achieve ToF measurements the RX must be synchronized to the TX. That is the RX needs to be aware of the clock offset between the two. This is important for the receiver to know when to expect the signal transmissions. In the absence of this

knowledge the clock offset directly impacts the measured ToF and can cause longer or shorter(even negative) ToF measurements. Therefore it is vital for functionality to mitigate the effects of this offset.

One can point to the lack of such worries for reflective ranging. The main reason for the presence of such an offset is the transmission and reception occurring and different nodes. This is unlike the case of reflective ranging, where signal is transmitted and reflected of an obstacle present at the range to be measured. There the reflected signal is transmitted and received at the same node. Therefore the time base(i.e. exact transmission time) for the received signal is available and the measured time equals 2*ToF. This form of ranging is used in radar systems and often in industrial ranging systems. [57]. However in sensor network ranging, the form factors of the sensors, to which the distances need to be measured, are small enough to not provide any significant reflection. Hence it is not possible to use reflective ranging for sensor networks and transmitter receiver clock offsets need to be dealt with explicitly.

Signals with Different Speeds

One way to achieve synchronization is resetting or calibrating the RX clock periodically before performing any real transmission is to occur. However the catch in this case is that the calibration signal needs to have a much shorter ToF than the signal transmitted for the actual range measurement. To this end signals with different speeds, such as radio signals and acoustic signals can be utilized. Figure 5.3 includes an illustration of this method. The faster radio signal synchronizes the RX and TX and is followed by the slower acoustic signal to measure the ToF. The key to the functionality of this scheme is the six orders of magnitude difference between the speeds of sound and light (340m/s vs. 300Mm/s).

This is a very common form of synchronization for ranging in sensor networks
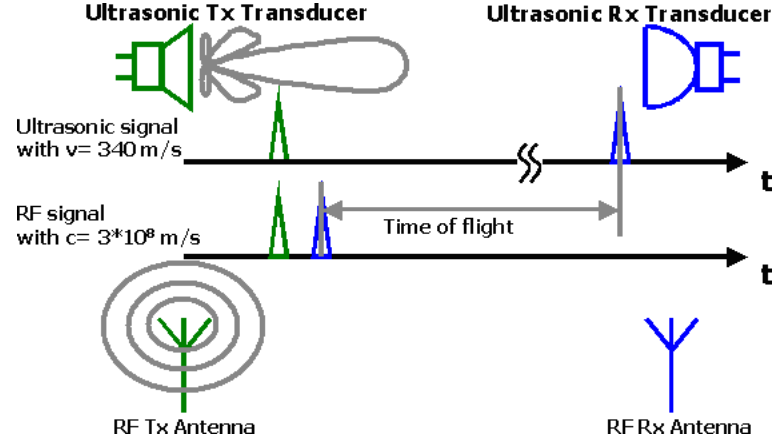
Figure 5.3: Illustration of synchronization using signals with different speeds

[45, 21, 58]. However, in addition to a radio transceiver these systems require highly directional, expensive and high power ultrasonic transducers [51]. Equipping each sensor node with such transducers would significantly increase the dollar cost as well as the power consumption of the sensor nodes.

Two Way Time Transfer

Another method to achieve synchronization is called two-way time transfer [59]. The main idea here is that unsynchronized ToF measurement results include the clock offset as an additive (subtractive) term. If the same measurement were carried out in the reverse direction, then the same clock offset(OS) would appear as a subtractive (additive) term. Averaging the forward and reverse measurement results cancels the contribution of the clock offset as shown in Figure 5.4. Also, halving the difference of the forward and reverse ToF measurements yield the clock offset between the TX and RX.

Determining the strongest tap at each channel estimate is called the time of arrival (ToA). If the clock offset between node A and node B is assumed to be A leading
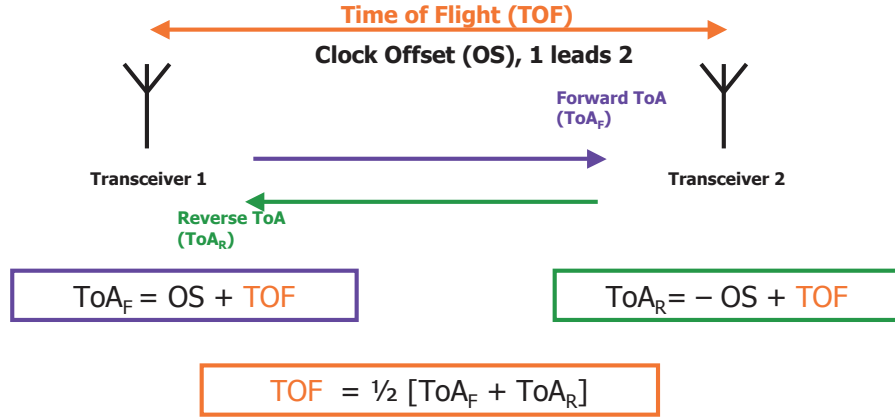
Figure 5.4:   Illustration   of   the   two   way   time   transfer   method   for   TX/RX synchronization

B by OS cycles. Also the ToF designates the time of flight between nodes A and B. Then the estimated ToA for the forward and reverse channels is computed using the following equations. Then the average of forward and reverse ToA's would yield the ToF. Also halving the difference of the ToAs would allow computation of the offset.   Associating the measurements with their particular time base offsets is a useful practise if these measurements are used for calibration purposes or any other purpose. It can also allow keeping track of the offset drift speed.

$$ToA_F = ToF + OS \tag{5.16}$$

$$ToA_R = ToF - OS \tag{5.17}$$

$$ToF = \tfrac{1}{2}[ToA_F + ToA_R] \tag{5.18}$$

$$OS = \tfrac{1}{2}[ToA_F - ToA_R] \tag{5.19}$$

There are two critical points should be noted regarding an implementation with two-way time transfer method. First, the clock offset needs to stay constant during the forward and reverse transmissions. Typically this implies that the measurements should be carried out in rapid succession. This clock offset drift is a direct result of the

different crystal oscillation frequencies at the each of the nodes. To avoid corruption of the results due to this frequency offset transmissions should be performed as rapidly as possible. Also a small frequency offset would also reduce the drift. Once the length of the data exchange sequence is determined, appropriate crystal accuracies can be calculated so that the clock offset will not drift within the measurement.

Last but not least, there needs to be an additional, reliable communication mechanism for exchanging transmit and receive times recorded at the other end of the link. In this way, the transmit and receive time of arrivals can be brought together and merged to yield the ToF and OS. Nevertheless this exchange is not timing critical and can be performed at a low data rate once the received data is sampled and the CIR as well as the time of arrival is obtained. In sensor networks, low data rate radios are already available for data communication [6], so such exchange is readily possible.

5.3 Conclusion

In this chapter ranging for sensor network localization has been systematically studied. The chapter started with an classification of ranging methods in terms of how they view the wireless channel. Here ranging methods using the timing view of the channel were found to be appropriate for their robustness against multipath effects. Next shortcomings with this view was addressed. Fundamental performance limits were investigated, OFDM based signaling was selected for its low computational complexity and two way time transfer was employed for TX/RX clock synchronization. After these selections in the next chapter the ranging system will be proposed.

Chapter 6

Ranging System

In the previous chapter the time view of the channel is chosen for implementing the ranging system and a time of flight measurement system was decided for implementation. Moreover the issues of the time domain view are addressed by deciding on using a multi carrier based signalling as well as two way time transfer for TX/RX synchronization. With these design decisions in place the ranging system can be designed.

6.1   Algorithm

To implement the ranging system the necessary functionality is a channel estimator. Channel estimation is an old problem that has been considered in many kinds of communication problems [41, 42, 43]. When transmitted data is comprised of only pilot symbols the optimal estimation method is matched filtering, may it be in time or frequency domain.

When an input signal X is transmitted, the received signal Y is the input filtered

through the channel.

$$y[n] = \sum_{i=-\infty}^{\infty} x[n-i] \ h_{channel}[i] \tag{6.1}$$

$$Y(\omega) = X(\omega) \ H_{channel}(\omega) \tag{6.2}$$

where, $y[n]$, $x[n]$ and $h_{channel}[n]$ are the time domain representations of the received signal, transmitted signal and CIR respectively. Also the frequency domain version of these signals are denoted as $Y(\omega), X(\omega)$ and $H_{channel}(\omega)$.

To find the channel frequency response $H_{channel}(\omega)$, the received signal frequency response can be divided by the transmitted signal's frequency response.

$$H_{channel}(\omega) = \frac{Y(\omega)}{X(\omega)} \tag{6.3}$$

Once $H_{channel}(\omega)$ is determined its inverse Fourier transform can be computed to obtain the CIR $h_{channel}[n]$. *This operation is frequency domain matched filtering.* Once the CIR is obtained, the time of arrival can be determined by finding the CIR tap with the largest absolute value.

As was stated earlier in the discussion on Two way time transfer synchronization and illustrated in Figure 5.4 the measured Time of arrival (labeled ToF$_F$) includes the TX-RX clock offset. A second time of arrival measurement in the reverse direction (labeled ToF$_R$) would include the same offset with the opposite sign, assuming the offset is constant throughout the measurement. Then the time of arrivals in the forward and reverse direction are averaged to calculate the ToF.

The following flow of events summarize the flow of the algorithm. In the transmit end, the pilot data is loaded in frequency domain, this pilot data is transformed to time domain. Digital pilot data in time domain is converted to continuous time using a Digital to Analog Converter(DAC). A Radio frequency (RF) front end up converts the DAC output to RF frequencies and transmits it with an antenna.
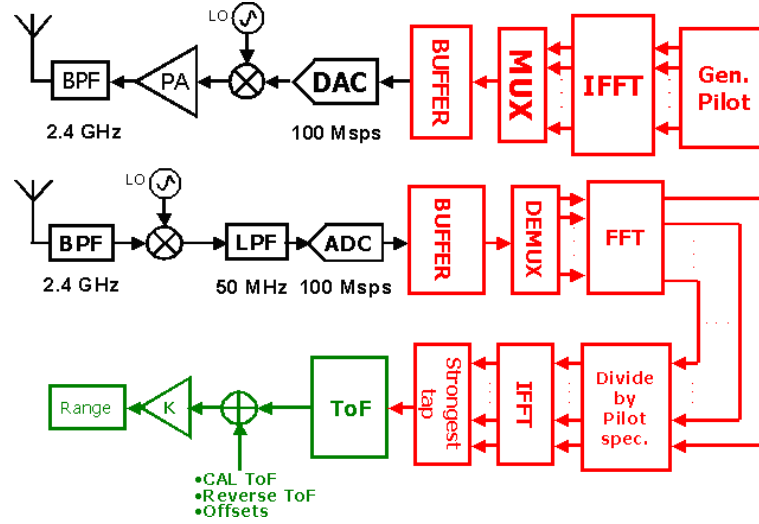
Figure 6.1: Block diagram of the ToF measurement ranging system.Upper chain is the TX part whereas the lower chain is the RX section.

At the receiver end the antenna captures the signal. The RF front end this time down converts the signal to baseband. The signal is digitized and buffered. Buffering effectively allows analog and digital processing to be carried out at different rates. This way the digital baseband operations can be computed at a clock rate much lower than the ADC sampling rate. The signal stored into the buffer consists of multiple periods of transmitted OFDM symbol all of which can be averaged to improve the signal to noise ratio (SNR) of the received signal.

Subsequently an FFT module transforms this digitized and buffered baseband signal into frequency domain. The signal frequency response is divided by the pilot frequency response yielding the channel impulse response. The division is carried out on an element by element basis. The FFT block is again used to execute an IFFT and transform the channel frequency response into CIR.

Here it should be pointed out that the RF local oscillator offset between the RX and TX is important, as it modulates the RX output at baseband. However

this modulation appears as complex gain $e^{(j2\pi f_{offset}t)}$ [41] in the CIR and does not affect the magnitude of the CIR taps. Hence the large CIR taps can be determined by observing their absolute values. Therefore as the last step a maximum search algorithm is employed to find the peak tap of the CIR absolute value and the index of this tap is assigned as the time of arrival.

A block diagram proposing a system to implement this algorithm is presented in Figure 6.1. Here the blocks shown on the right side of the buffer represent the digital baseband functions that can be implemented in digital circuits using an FPGA or ASIC. The analog front end blocks including the A/D and D/A converters are represented on the left side of the buffer. Last, the final ToF computations, which are simple arithmetic operations, as well as the associated data exchanges are shown after the *Strongest Tap* block in Figure 6.1 and can simply be implemented using a microcontroller or any similar programmable device.

6.2 Signal Design

To implement the algorithm described in the previous section a suitable signal needs to be designed. There are a number of signal parameters that are important for various aspects of the ranging system. These parameters include the sampling rate used in the system, bandwidth of the used signal, number of carriers in the multi carrier signal and the bitwidths in the digital baseband.

6.2.1 Signal Bandwidth

Signal bandwith is an important parameter that affect ranging accuracy. As was explained out in Section 5.2.1 the CRLB of the ranging error variance is improved with increasing signal bandwidth. However this benefit does not come for free. Increasing

the signal bandwidth without bounds would cause two significant problems. First, the required sampling rate to capture the signal in digital domain would increase increasing the power consumption due to the required analog to digital (A/D) and digital to analog (D/A) converters. Secondly finding available RF spectrum would be difficult. As the signal bandwidth exceeds 100MHz the available bands would either require very low power transmission [53] or operation within unlicensed bands at high RF frequencies.

To have a quick sense of the CRLB vs bandwidth(f) relation at a moderate 10 dB SNR value, Equation 5.10 can be used. Replacing F_s in this equation with $2f$ yields

$$\sigma_{range} = \frac{15.23}{f}(m/MHz) \tag{6.4}$$

At this signal level to achieve an error deviation less than 1m the requires that the signal bandwidth is

$$f = \frac{15.23}{\sigma_{range}}[MHz] = \frac{15.23}{1} = 15.23MHz \tag{6.5}$$

therefore the CRLB predicts that at 10dB SNR a signal with 15.23MHz bandwidth is necessary to achieve ranging error deviation less than 1m. For lower SNR values the CRLB would be higher meaning the error variation would be larger. Repeating the computations that led to Equation 5.10 at a low SNR value of 3 dB, the measured range deviation σ_{range} vs signal bandwidth f can be derived as

$$\sigma_{range} \approx \frac{42}{f}[m/MHz] \tag{6.6}$$

to achieve $\sigma_{range} = 1m$

$$f = \frac{42}{\sigma_{range}}[MHz] = 42MHz \tag{6.7}$$

The derivation of Equation 6.6 is included in Appendix Chapter A.

The important point is that the CRLB of the range measurements impose signal bandwidth requirements. However it should be kept in mind that the ToF quantization effects caused by sampling can prevent to approach the CRLB. Therefore before settling on the signal the accuracy requirement due to the sampling should be taken into consideration.

6.2.2 Sampling rate

In Section 5.2.1 the sampling rate was determined to have an effect on the ranging accuracy that may be more limiting than the CRLB in case of Nyquist sampling. Hence, the bounds imposed by the sampling rate also needs to be adjusted such that they remain within the desired performance.

It was proposed in Chapter 1 that ± 1m ranging error goal is a realistic target that can be translated to a ± 0.5m of maximum position error by using overconstrained localization algorithms. To meet the ranging accuracy goal, the result of Equation 5.7 can be utilized. Rearranging this equation yields

$$F_s = e_{range} * 150[Msps] = 150Msps \tag{6.8}$$

Hence if $e_{range} = 1m$ then the the OFDM signal should be sampled at 150Msps. As discussed in previous section with similar specifications the CRLB at high SNR requires a $15MHz$ signal whereas the CRLB at low SNR requires a $42MHz$ signal. The respective Nyquist rates associated with these signals are $30Msps$ and $84Msps$. That is the sampling rates necessitated by the CRLB bound are lower than the sampling rates necessitated by the ToF quantization.

In the proposed system txhe accuracy requirement should be satisfied by setting the sampling rate at 150Msps and the signal bandwidth at 75MHz in Nyquist sampling. However, as will be discussed in Chapter 8, the in the prototype system, ADC

and DAC only allow operation speeds up to 100Msps and the maximum signal band-
width is limited to 50MHz. Therefore this was the sampling rate and signal bandwidth
used in the proposed system. It should be noted that with 100Msps sampling rate
the maximum ranging error becomes $e_range = \pm 1.5m$ and the error deviation (using
Equations 6.4 and 6.6) becomes 0.8m and 0.3m for low SNR and high SNR settings.

With a 50MHz bandwidth, the system can fit in the 2.4-2.5GHz ISM band. This
enables the use of existing commercial analog parts designed to work in this band for
system prototyping. Additionally, the relatively high power levels allowed in this ISM
band help to have enough signal strength albeit at the cost of higher interference.

At this stage it is important to distinguish that the sampling does not need to
be at the Nyquist rate. That is the signal bandwidth does not necessarily have to
be at half the sampling rate. It is also possible that the signal bandwidth be set
such that CRLB would be lower than a requirement and then this signal can be
over sampled such that quantization error due to sampling effects would be within
acceptable limits. Even though this fact does not make any difference for our case as
with 100Msps sampling the main performance bottleneck is the sampling rate it can
allow operation narrower bandwidths given availability of SNR.

6.2.3   Number of carriers

After the signal bandwidth and sampling rate has been decided, decisions regard-
ing the characteristics of the multi carrier signal itself should be made. The most
important property of this signal is the number of its subcarriers. The number of
OFDM carriers affect the performance of the ranging system in a number of ways.

The upper limit of the number of carriers is determined by Local oscillator(LO)
phase noise and the peak to average ratio (PAR). The lower limit is set by the channel
impulse response delay spread.
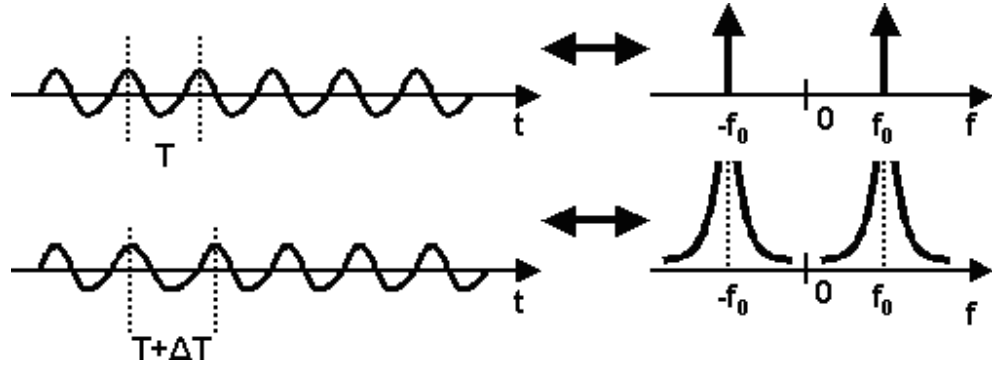
Figure 6.2: Upper figure shows a noiseless sinusoid. The lower figure shows the effect of phase noise in time and frequency domains

Phase noise

"In a real oscillator operating at a frequency f_0 circuit and system noise varies the period of oscillation randomly-as if the oscillator also operates at frequencies other than f_0." [60] Here the changing frequency implies varying instantaneous frequencies and the spectrum is not a delta function at the operation frequency(f_0) but rather a continuum of frequencies around this point. This effect is illustrated in figure 6.2 the skirts around the carrier frequency represent the smeared single frequency tone. When plotted on log-log axes the skirts often show to have a single pole low pass response (20dB/dec rollof) and they therefore they are usually specified with their -3dB corner frequencies.

Since phase noise has a broadening effect on the spectrum, after the baseband signal is up converted to the RF bands by using a noisy Local Oscillation signal each of the OFDM subcarriers are broadened in frequency domain. Therefore if the OFDM subcarriers are packed to closely to each other the skirts start interfering more with each other. This is equivalent to Inter Carrier Interference (ICI) in multicarrier systems. Presence of ICI degrades the Orthogonality of the subcarriers and makes FFT based demodulation difficult [54]. Therefore increasing the number of carriers

without bound would incur an increasing ICI and degrade performance. Moreover the amount of SNR degradation due to ICI can be calculated as [61]

$$D_{phase} \cong \frac{11}{6ln10} 4\pi \ \beta T \frac{E_s}{N_0} \qquad (6.9)$$

where, β is the -3dB corner frequency of the phase noise spectrum shape and T is the OFDM symbol period with $1/T$ as the subcarrier spacing. Using this equation, it can be seen that SNR degradation is proportional to

$$D_{phase} \propto \beta T = \frac{\beta}{\Delta f} = \frac{\beta}{2f_{BW}/N} = \frac{\beta N}{2f_{BW}} \qquad (6.10)$$

where, Δf is the subcarrier spacing, N is the total number of sub carriers and f_{BW} is the baseband signal bandwidth. As can be seen the phase noise is directly proportional to N the number of carriers. Assuming a moderate -70dBc phase noise at 500kHz offset the corner frequency would be

$$\beta \approx 10^{\frac{-70}{20}} \times 500k = 158Hz \qquad (6.11)$$

Next combining Equations 6.9, 6.10 and 6.11 to obtain a 3dB(or 50%) SNR degradation

$$D_{phase} = 0.5 = \frac{11}{6ln10} 4\pi \ \frac{\beta N}{2f_{BW}} \qquad (6.12)$$

$$0.05 = \frac{\beta N}{2f_{BW}} = \frac{150N}{100MHz} \qquad (6.13)$$

$$N = 33k \approx 2^{15} \qquad (6.14)$$

therefore it can be concluded the upper limit of number of carriers due to phase noise is not a real limitation.

Peak to average ratio

Peak to average ratio is a significant problem for multicarrier systems [41, 54]. The problems stems from the fact that when transmitting multicarrier signals such

as:

$$x(t) = \frac{1}{N} \sum_{n=0}^{N-1} a_n e^{j2\pi f_n t} \tag{6.15}$$

The peak power would be

$$P_{peak} \propto N^2 P_b \tag{6.16}$$

whereas the average power would be

$$P_{avg} \propto N P_b \tag{6.17}$$

and finally

$$PAR \propto \frac{P_{peak}}{P_{avg}} = N \tag{6.18}$$

where, a_n are the transmitted symbols, N is the number of carriers, P_{peak} is the peak OFDM symbol power, P_{avg} is the average power of OFDM symbols, PAR is the Peak to Average ratio and P_b is the bit power Therefore increasing N would increase the PAR. Higher PAR usually translates to difficulties in A/D as well as D/A conversions in addition to maintaining power amplifier (PA) linearity and efficiency [54].

Since the ranging subsystem does not need to transmit random data the transmitted pilot signals can be selected such that the PAR problem can be mitigated. Research on such signal encodings already exist in literature [41, 54]. Even though such remedies provide limited relief in case of random signal transmissions the completely deterministic pilot signals used in the ranging subsystem can reduce the problems caused by PAR issues.

CIR Delay spread

As mentioned earlier a low limit on the number of carriers is necessitated due to the CIR delay spread. Under normal operation conditions it is necessary that the channel estimate includes all the channel taps that has significant energy.

Violation of this condition the CIR would have aliasing and the measurements would be corrupt. The delay spread of different channels have different values depending on the size and other physical properties of the channel. When using an N point OFDM signal as the multicarrier signal. The CIR would have N samples. These N samples would represent the time in the range of [-N/2, N/2 - 1] and only half of them can be utilized to represent the delays which would occur at positive indexes of the CIR estimate. So using N subcarriers and a sampling period of T_s, a maximum delay of (T_{max}) can be represented as follows.

$$T_{max} = \frac{N}{2} * T_s \qquad (6.19)$$

For a sensor network application it was earlier stated that maximum internodal distances are expected to be around 10–15m and the network size is expected to be around 100m × 100m. Admittedly even though these maximum ranges and network dimensions assumed in this design are increasing as the sensor networks themselves evolve [49].

However using these two numbers as maximum radio range and network size and the speed of light ($c = 30$cm/ns) the LoS ToF is expected to be upper limited by 33–50ns. In addition the ToF within the network would be upper limited by 330–500ns. hence it can be estimated that with such a network size the maximum expected delay would occur within 500ns. Therefore using this value with a sampling period of 10ns in Equation 6.19

$$N = \frac{2T_{max}}{T_s} = \frac{2 \times 500ns}{10ns} = 100 \qquad (6.20)$$

Therefore the Number of Carriers should be more than 100 for it to be able to accommodate the delay spread of the possible wireless channel. For the work presented in this thesis a 128 point FFT was selected for implementation. Hence the

System Parameter	Value
Sampling frequency (F_s)	100Msps
Signal bandwidth (f_{BW})	50MHz
Number of Carriers	128
Symbol period	1.28μs
Subcarrier spacing (Δf)	766kHz
ADC resolution	6 bits
Frequency band	2.4GHz

Table 6.1: Summary of the proposed system specifications

OFDM signal has 128 carriers and at 100Msps, the OFDM symbol period is 1.28ms.

There exists another interpretation of looking at the dependence of delay spread that can be captured with the CIR estimate on the number of carriers. Here, assuming the transmission and reception occurs during one OFDM symbol, all the channel taps including LoS and all other multipath arrivals should arrive during one OFDM symbol. Otherwise this would cause multipath arrivals in the next OFDM symbol causing Inter Symbol Interference (ISI).

An important consequence of the symbol length is the total duration that it will take to complete the measurements. If four OFDM symbols are received during each acquisition it will take 5.12μs to finish acquisition in one direction. Assuming it takes 20μs to handshake and overhead for reception a total of 25μs are needed for a one way ToF measurement. Finally the two way ranging can finish in 50μs.

6.3 System simulations

After determining the algorithm and key system parameters that are necessary for the system implementation, these values are summarized in Table 6.1. Next, initial system analysis using these specifications needs to be carried out using computer
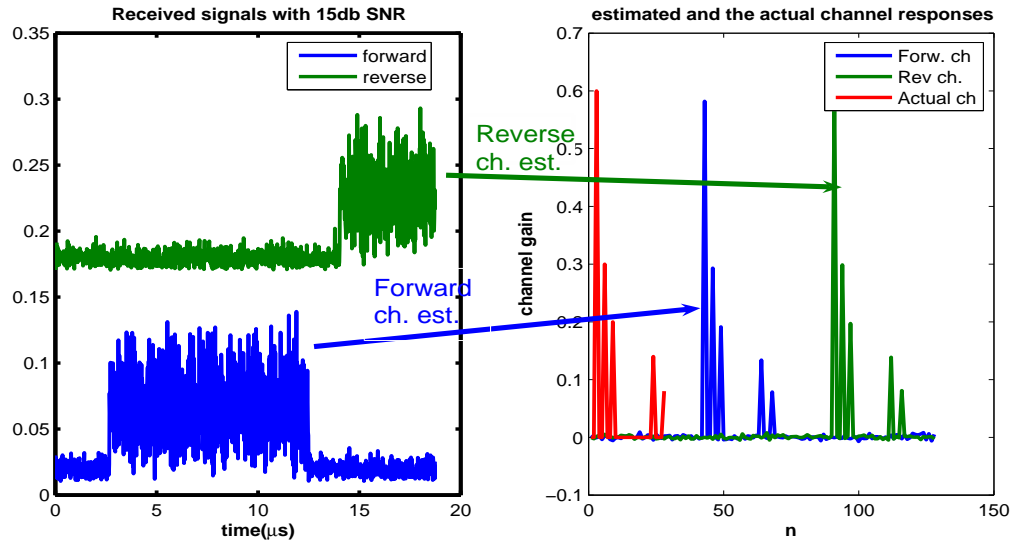
Figure 6.3: Simulated waveforms during forward and reverse transmissions (Left), Estimated reverse, forward and the actual channels. (Right)

simulations. The goal of these simulations are twofold: Ensuring the algorithm functionality in an environment with limited and controlled nonidealities. Observing the effects of a number of selected nonidealities.

The system simulations were carried out in Matlab. The simulated channel was a 5-tap Rician channel. The channel had an exponential delay profile and a 100ns delay spread. The range was modeled as the delay of the first channel tap. To modify range, and therefore modeled delay, the first tap delay was modified.

Specifically the channel was generated at a much higher sampling rate than the sampling rate of the target system. For example during simulations this high rate was 1Gsps. At this rate the range is modeled as the delay of the first tap and this delay is also added to all the remaining taps so that the channel delay profile would be preserved. Then this channel is downsampled to the signal sampling rate of 100Msps. This downsampling is performed using the `resample` function in Matlab. Using the downsampled CIR the system is simulated. The transmitter and receiver clock offset
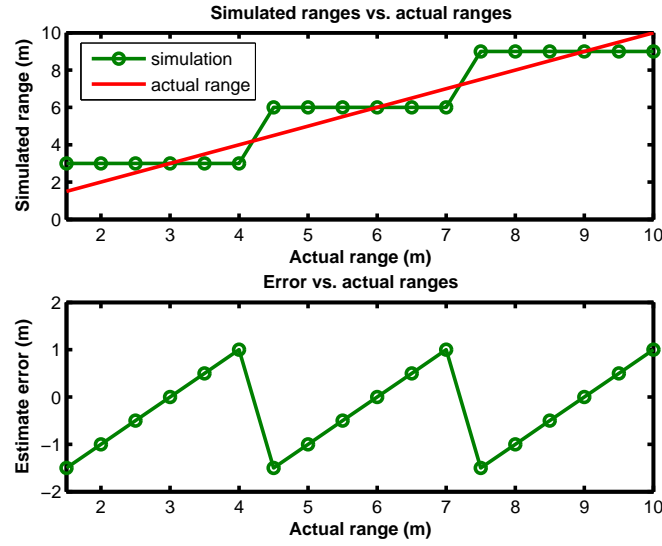
Figure 6.4: Estimates and errors obtained as a result of simulations

is modeled as a time shift in the OFDM symbol beginnings in the transmitter and the receiver.

The system simulations are carried on strictly on the baseband equivalent of the ranging system at the baseband sampling rate. The RF front end and its effects are not included in simulations except for RF front end effects that have baseband equivalents. Most significant of these effects is the LO phase noise.

Another effect modeled in the simulations are the limited resolution of the input signals. The input signals are saturated and quantized to a 6 bit level. This effect is realized using the rounding the full floating precision point inputs to their quantized values. However during the rest of processing the datapath blocks are assumed to have floating point precision. Last modeled effect is the contribution of the LO phase noise to the OFDM baseband signal. This phase noise is modeled to have a first order response with a -70dBc gain at 500kHz.

Figure 6.3 shows the progress of simulations. Here the left side graph shows the

simulated received waveforms whereas the right side shows the estimated CIR signals. On the left side figure, the lower waveform represents the forward transmission algorithm. The forward transmission lasts from 2.5 to $12.5\mu s$ for $10\mu s$. During this time the signal is transmitted from a node A and is received and processed at node B. It is also at this node a forward channel estimate is obtained.

Immediately after the forward signal transmission terminates a reverse signal transmission commences this time from node B to node A at around $15\mu s$. Using the data acquired at node A the reverse channel is estimated. In the right side of Figure 6.3 these forward and reverse channel estimates and depicted. Note that in these simulations the possible wasted time that can be spent during handshaking and arbitrating turns of transmission.

Figure 6.4 shows the system simulation results as the range is swept between 1.5m and 10m. It is seen from the figure that the ranging error varies between $[-1.5\text{m}, 1\text{m}]$ interval. The simulated system can distinguish ranging increments of 3m or more as expected.

Last but not least, the input signal used for the simulations is a 128 point Shapiro-Rudin sequence [54] that is intended to yield a low PAR ration in the OFDM baseband signal. Even though this signal eventually was not implemented for the prototype for various reasons, as expressed in the previous sections using such a signal during final operation achieves lower PAR levels.

6.4   Conclusion

In this chapter the relevant parameters of the ranging system are decided. The implemented system uses a 128-pt muli carrier system that is sampled at 100Msps. The signal bandwidth is 50MHz receive ADC has 6bits of resolution. It fits and

operates in the 2.4 GHz ISM band. The system has been simulated in Matlab for performance evaluations. As a result the algorithm is found to be functional with ranging error in the range of $[-1.5\text{m}, 1\text{m}]$. Now that the system is designed and all parameters are selected. The next step is to realize its digital baseband. This is the topic for the next chapter.

Chapter 7

Digital baseband implementation

Once the system design is in place and initial system simulations confirm its performance with the used set of parameters. The next task is implementing pieces of the localization system. In this project, the application specific functions of the ranging block are its digital baseband functions. These functions include the tasks performed by the red blocks in Figure 6.1 and include pilot generation, IFFT/FFT, buffer, division, peak searching etc.

The system description that was conceived section 6.1 and simulated in Matlab is hand converted to an RTL level design. The RTL language of selection is VHDL (VHSIC Hardware Description Language). During the course of the project the system RTL description is first synthesized with an FPGA library so that the system can be prototyped, as will be described in the next section. Second, the design is synthesized with an ASIC library for a final IC implementation. For the two platforms there are some differences in their respective RTL descriptions. However these differences are kept at minimum by effective use of hierarchy. In the rest of this section, the implementation of significant subblocks for the FPGA as well as the ASIC implementations are presented.

7.1 FFT/IFFT unit

FFT is a computationally efficient method to implement discrete fourier transforms (DFT) and compute frequency domain representations of time domain signals. For a time domain signal $x[n]$ its N-point DFT is mathematically formulated as

$$X[k] = \sum_{n=0}^{N-1} x[n] \, W_N^{nk}, \qquad\qquad k = 0, 1, \ldots, N-1 \qquad (7.1)$$

$$W_N^{kn} = e^{-j(2\pi/N)kn} \qquad (7.2)$$

There are two main alternatives in implementing such a transform.

- In the first implementation the time domain signal $x[n]$ is separated into even and odd indexed samples. This method is called decimation in time (DIT).

- In the second implementation even and odd terms of the frequency domain representation $X[k]$ are separately computed. Since the separation is in frequency domain this method is called as decimation in frequency (DIF) [62].

7.1.1 Background on FFT algorithms

Next decimation in time and frequency algorithms can be compared with respect to their implementation complexities. The following analyses and derivations follows closely to that presented in Oppenheim and Schafer [62] and included here for the sake of completeness. First considered below is the decimation in time implementation of
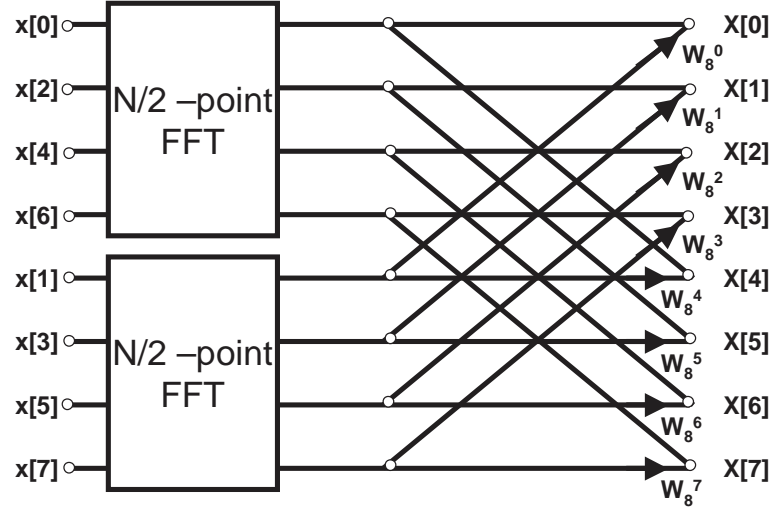
Figure 7.1: Flow graph illustrating the decimation in time (DIT) method to bring two N/2 point FFTs to obtain an N point FFT

the FFT algorithm.

$$X[k] = \sum_{n\ even}^{N-1} x[n]\ W_N^{nk} + \sum_{n\ odd}^{N-1} x[n]\ W_N^{nk} \tag{7.3}$$

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r]\ W_N^{2rk} + \sum_{r=0}^{(N/2)-1} x[2r+1]\ W_N^{(2r+1)k} \tag{7.4}$$

$$= \sum_{r=0}^{(N/2)-1} x[2r]W_{N/2}^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x[2r+1]W_{N/2}^{rk} \tag{7.5}$$

$$= G[k] + W_N^k H[k] \tag{7.6}$$

Each of sums in Equation 7.6 is recognized as an N/2 pt FFT by its own [62]. Hence the N point FFT is reduced to 2 N/2 FFTs followed by a proper way to combine them. Figure 7.1, adapted from [62], illustrates this idea for an 8 point FFT.

On the other hand instead of separating the time domain samples with respect to their time domain indexes the frequency domain samples can be calculated in two groups depending on their indexes. This method is called decimation in frequency.

In this case, again following [62]

$$X[k] = \sum_{n=0}^{N-1} x[n]\ W_N^{nk}, \quad k = 0, 1, \ldots, N - 1 \tag{7.7}$$

$$X[2r] = \sum_{n=0}^{N-1} x[n]W_N^{n(2r)}, \quad r = 0, 1, \ldots, (N/2) - 1 \tag{7.8}$$

$$X[2r] = \sum_{n=0}^{(N/2)-1} x[n]W_N^{2nr} + \sum_{n=N/2}^{N-1} x[n]W_N^{2nr} \tag{7.9}$$

$$X[2r] = \sum_{n=0}^{(N/2)-1} x[n]W_N^{2nr} + \sum_{n=0}^{(N/2)-1} x[n+N/2]W_N^{2(n+N/2)r} \tag{7.10}$$

However noting periodicity of W_N^r with N

$$W_N^{2(n+N/2)r} = W_N^{2nr+Nr} = W_N^{2nr} \tag{7.11}$$

$$X[2r] = \sum_{n=0}^{(N/2)-1} (x[n] + x[n+(N/2)])W_{N/2}^{nr} \tag{7.12}$$

Hence the even indexed elements of the frequency spectrum can be obtained by adding the first half and last half of input sequence and taking the N/2 FFT of the result [62]. Next the odd indexed entries of the frequency response can be computed as follows.

$$X[2r+1] = \sum_{n=0}^{N-1} x[n]W_N^{n(2r+1)}, \quad r = 0, 1, \ldots, (N/2) - 1 \tag{7.13}$$

$$X[2r+1] = \sum_{n=0}^{(N/2)-1} x[n]W_N^{(2r+1)n} + \sum_{n=N/2}^{N-1} x[n]W_N^{(2r+1)n} \tag{7.14}$$

$$X[2r+1] = \sum_{n=0}^{(N/2)-1} x[n]W_N^{(2r+1)n} + \sum_{n=0}^{(N/2)-1} x[n+N/2]W_N^{(2r+1)(n+N/2)} \tag{7.15}$$

The factor in the second summation can be expanded as

$$W_N^{(2r+1)(n+N/2)} = W_N^{(2r+1)n}W_N^{rN}W_N^{N/2} = W_N^{(2r+1)n} \times 1 \times -1 = -W_N^{(2r+1)n} \tag{7.16}$$
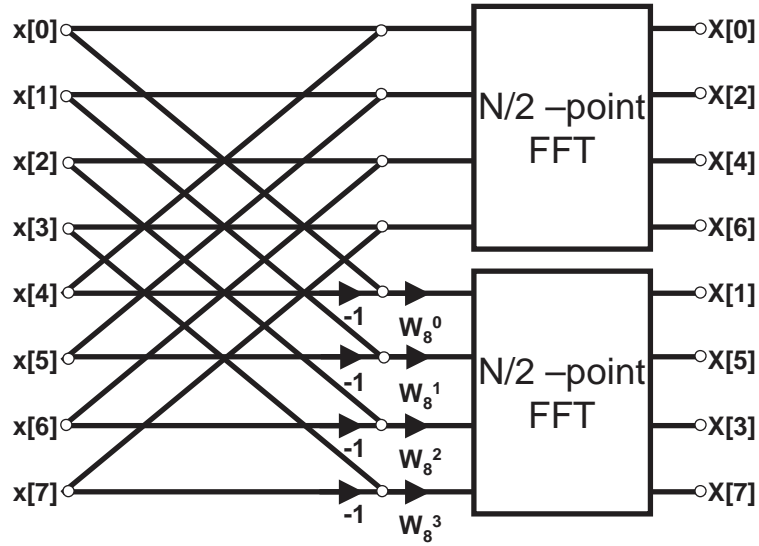
Figure 7.2: Flow graph illustrating the decimation in frequency (DIF) method to bring two N/2 point FFTs to obtain an N point FFT

and finally

$$X[2r+1] = \sum_{n=0}^{(N/2)-1} (x[n] - x[n + N/2])W_N^{(2r+1)n} \tag{7.17}$$

$$X[2r+1] = \sum_{n=0}^{(N/2)-1} (x[n] - x[n + N/2])W_N^n W_{N/2}^{rn} \tag{7.18}$$

From this derivation it is concluded that the odd terms of the frequency points can be obtained as the N/2 point FFT of the sequence obtained by $[x(n) - x(n + N/2)]W_N^n$ for $0 \le n < (N/2)$ whereas its even terms are obtained by computing the N/2 point FFT with the terms $[x(n) + x(n + N/2)]$. This is illustrated in Figure 7.2

As illustrated above both decimation in time (DIT) and decimation in frequency (DIF) approaches allow computation of an N point FFT by using two N/2 point FFTs. This idea can be further pursued by computing the N/2 point FFTs from N/4 point FFTs and such. Below, this algorithm is realized for a small 8-pt FFT. In the first stage the 8-pt FFT is reduced to two 4-pt FFTs, which are in turn reduced to
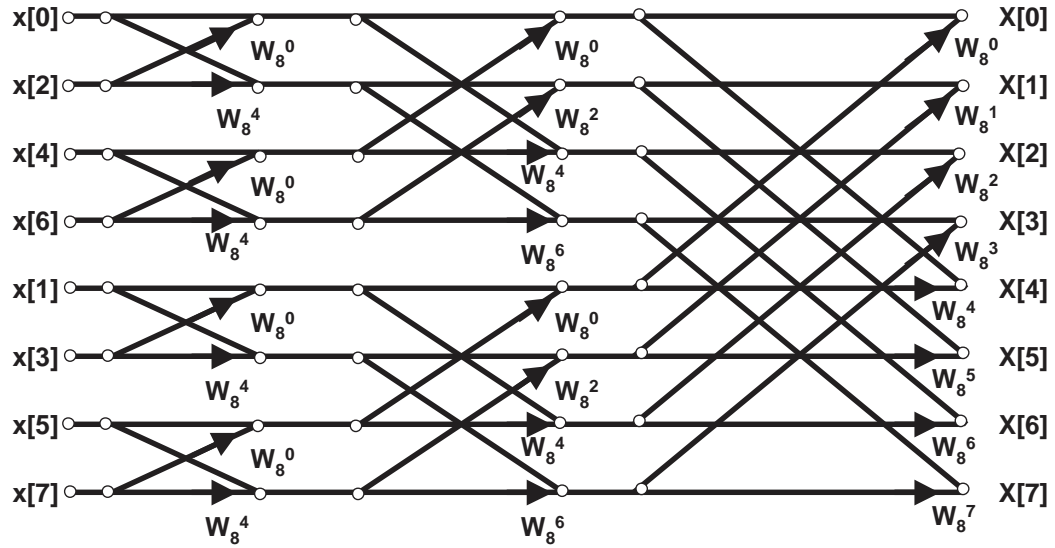
Figure 7.3: Flow graph for a complete FFT computation implemented in decimation in time

four 2-pt FFTs in the next stage. 8-pt FFT implementations using DIT and DIF approaches are illustrated in Figures 7.3 and 7.4 respectively.

As seen in these figures at each stage of the FFT there are operations either combining the results from the previous stage of smaller FFTs (as in DIT) or combining and preparing the data for the smaller subsequent FFTs (as in DIF). For each of these computation stages the data out of or in to the $N/2$ -pt FFT stage the elements indexed i^{th} and $(i + N/2)^{\text{th}}$ are combined to yield two outputs. That is the data is processed in pairs using a basic kernel computation. Due to the shape of its associated flowgraph this operation is termed a butterfly computation. Figure 7.5 illustrates these kernel computations in the case of DIT and DIF based FFT algorithms.

The corresponding formulations of these butterfly computations are quite straight-
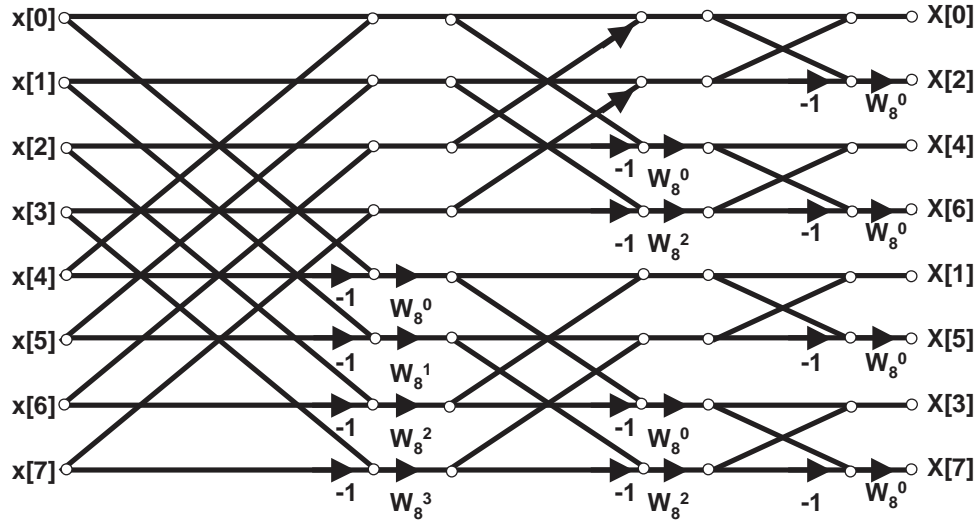
Figure 7.4: Full FFT implementing decimation in frequency
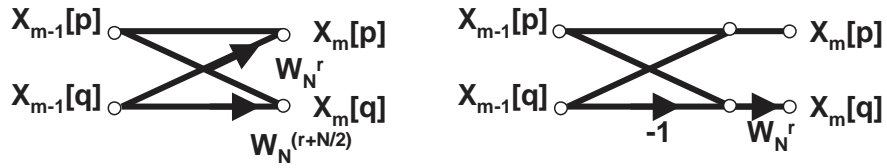
Figure 7.5: Flowgraphs for Decimation in time butterfly operation (left) and Decimation in frequency operation (right)

forward. The DIT basic butterfly can be formulated as

$$X_m[p] = X_{m-1}[p] + W_N^r X_{m-1}[q] \tag{7.19}$$

$$X_m[q] = X_{m-1}[p] + W_N^{r+N/2} X_{m-1}[q] = X_{m-1}[p] - W_N^r X_{m-1}[q] \tag{7.20}$$

Whereas the DIF butterfly can be formulated as

$$X_m[p] = X_{m-1}[p] + X_{m-1}[q] \tag{7.21}$$

$$X_m[q] = (X_{m-1}[p] - X_{m-1}[q])W_N^r \tag{7.22}$$

From the flowgraphs it is apparent that at each stage of the FFT computation, one data point is associated with only one butterfly computation. This property allows an in place computation of the FFT. Additionally it is observed that the indexing of the input and output points are unconventional. However it is easy to note that when written in binary format the input and output indexes are just bit-reversed versions of each other.

$$x[000] \leftrightarrow X[000]$$
$$x[100] \leftrightarrow X[001]$$
$$x[010] \leftrightarrow X[010]$$
$$\vdots$$
$$x[011] \leftrightarrow X[110]$$
$$x[111] \leftrightarrow X[111]$$

Also observed from the flow diagrams is that each butterfly operation is associated with a complex multiplication or rotation factor sometimes called a "twiddle" factor. The value of this coefficient also depends on the FFT stage and index of the butterfly inputs.
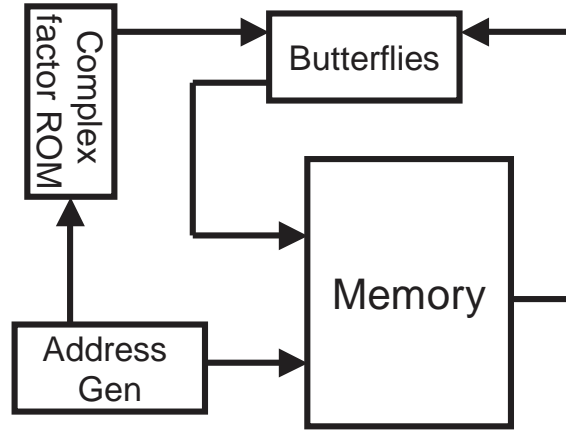
Figure 7.6: Block diagram of the FFT unit

7.1.2   FFT unit realization

Theoretically the DIT and DIF approaches to the FFT computation are equivalent to each other. The implemented FFT unit was realized using the DIF approach for computing the FFT. Even though there are no significant implementation differences either the reason for implementing this DIF implementation is it needs a complex multiplication for only one path of the butterfly operations. Also the here only the output of the subtractor is input to the multiplier whereas in the DIT approach the multiplier output is input to two adders.

When realizing the FFT unit the key blocks are the butterflies, memories including a complex factor ROM and an address generation logic unit. A crude block diagram of this FFT unit is shown in Figure 7.6.Note that the same block can be used to compute the inverse FFT by only negating the W_N^n coefficients.

First FFT key components are the butterflies. The basic flow diagram of these computations were introduced in the last section. Here butterfly modules were implemented using fixed point arithmetic as shown in Figure 7.7. In this figure the
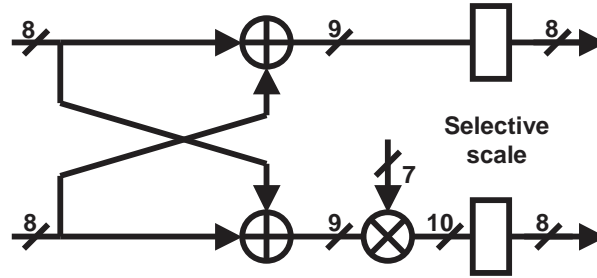
Figure 7.7: Implementation details of the butterfly computation

8-bit inputs are added first for 9 bit results. Whereas the even indexed output needs no further processing, the odd indexed output path needs to be multiplied by complex coefficients. These are represented by 7-bit complex numbers and the result is truncated down to 10-bit complex numbers. Finally before outputting the 9-bit and 10-bit numbers they are selectively scaled down. Selective If during any stage any one output of the N/2 butterfly operations have possibility of overflow, in the next FFT stage all the butterfly results are scaled by 2 to avoid overflow. If noneof he butterfly outputs are deemed possible to overflow in the next stage then there is no scaling. This procedure is called selective scaling.

Two butterfly units are employed to function in parallel and each FFT stage is completed in 32 computations. Due to use of synchronous or clocked memories read/write operation takes two cycles. Therefore one stage of the FFT computation is completed in 64 cycles.

Second key components of the FFT are the memory blocks. To begin with, memories are synchronous memories. That is the read values are available in the next cycle after address is changed. Therefore for a read-process-write operation two cycles are required. Using a single cycle read/write memory can cause race conditions and using synchronous memories avoids such problems. In this project, two dual

Data1[15:0]

Addr1[5:0]

Data2[15:0]

Addr2[5:0]

0-63

out1[15:0]

out2[15:0]

Data3[15:0]

Addr3[5:0]

Data4[15:0]

Addr4[5:0]

64-127

out3[15:0]

out4[15:0]
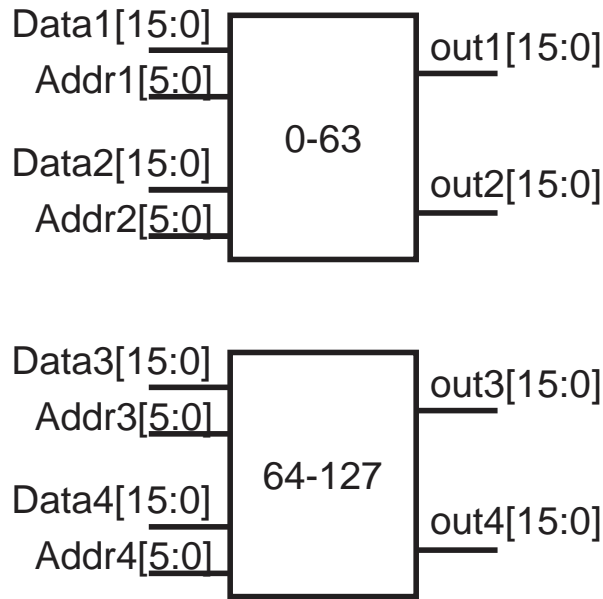
Figure 7.8: Block diagram of the memories of the FFT unit

ported memories with 64 16-bit words are used so that 128 8-bit complex numbers can be stored. The two memories represent the upper and lower halves of the overall memory space. The overall effect is a pseudo 4-port memory. That is 4 elements of data can be accessed at once but 2 of these are from the upper half while the other two is from the lower half of the memory. Since the data is accessed in an orderly fashion the pseudo 4-port property of the memory proves possible to utilize.

The original need for a 4 port memory arose from the need of quadrupling clock rate from the baseband sampling rate to the DAC input rate. In the prototype the on board clock rate is 25MHz whereas the maximum DAC and ADC sampling rates are 100Msps. In this case the need for using a 4 to 1 multiplexer required a memory that can output 4 data points at one clock cycle and can be clocked at the slow rate. Therefore designed was the 4 port memory clocked at the slow rate.

The additional piece of needed memory was due to the complex coefficients.

ind	1st	2nd	3rd	4th	5th	6th	7th
0	$0-64$	$0-32$	$0-16$	$0-8$	$0-4$	$0-2$	$0-1$
1	$1-65$	$1-33$	$1-17$	$1-9$	$1-5$	$1-3$	$2-3$
2	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$2-6$	$4-6$	$\vdots$
3	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$3-7$	$\vdots$	$\vdots$
4	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$8-12$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
7	$\vdots$	$\vdots$	$\vdots$	7–15	$\vdots$	$\vdots$	$\vdots$
8	$\vdots$	$\vdots$	$\vdots$	16–24	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
15	$\vdots$	$\vdots$	$15-31$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
16	$\vdots$	$\vdots$	$32-48$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
31	$31-95$	$31-63$	$47-63$	$55-63$	$59-63$	$61-63$	$62-63$
32	$32-96$	$64-96$	$64-80$	$64-72$	$64-68$	$64-66$	$64-65$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
63	$63-127$	$95-127$	$111-127$	$119-127$	$123-127$	$125-127$	$126-127$

Table 7.1: Butterfly input indices in decimal format

Rather than a read/write memory the need was for a ROM that can hold the complex coefficients that implement the complex factors W_N^n in Equation 7.18. Possibly of additional use was the negative values of these coefficients which would be needed for the case of an IFFT computation.

Third and last key component is the address generation. The function of this block is generating the RAM addresses such that appropriate butterfly inputs could be read out at each stage of the FFT. To compute an N-pt FFT at each stage stage N/2 butterfly computations are needed. For the DIF implementation, considering the pairing of the data for use in the butterflies, the indexes are N/2 apart for the first stage, N/4 for the second stage and $N/2^i$ for the ith stage. This requirement can be concluded by observing both from the sample 8-pt FFT flowgraph shown in Figure

7.4 and Equations 7.12 and 7.18

For the 128 point FFT that is designed for the ranging system butterfly input indexes, which are read out of the memory, would be 64 apart during the first stage, 32 in the second stage and so on and so forth. Hence the butterfly input indexes over FFT stages can be summarized as below in Table 7.1. This table summarizes the indices of the pairs that are processed together in the butterfly units throughout the FFT stages. The next table repeats the same pairs in 7-bit binary format so that patterns can be recognized for simple address generation.

From Table 7.2 it is clear that all the indices can be generated by a single 6-bit counter counting from 0-63. Each stage of the FFT can be computed in 64 cycles if a single butterfly is used. However since two butterfly units(A and B) are used and the used memory can allow access up to 4 data points, a 5-bit counter that counts up to 32 can be used to complete each FFT stage in 32 cycles.

Since the memory halves are 64 words each the memory addresses are 6-bits whereas the 7th index bit selects the output from the upper or lower memory half. That is the lower 6bits of the index are connected to the memory address whereas the 7th bit(MSB) is connected to the output selection MUX. Using these 7 bits the indices of inputs to the butterflies A and B can be tabulated as in Table 7.3. Here A_1, A_2 and B_1, B_2 represent the first and second inputs to butterflies A and B, W represents the address input to the complex coefficient ROM while c is the 5-bit counter output.

Once the FFT computation stages are over the results are read out in the bit reversed addressing. That is the counter bits are reversed and used as the address bits of the memory when reading the frequency domain representation of the input.

ind	1st	2nd	3rd	4th	5th	6th	7th
0	0000000 – 1000000	0000000 – 0100000	0000000 – 0010000	0000000 – 0001000	0000000 – 0000100	0000000 – 0000010	0000000 – 0000001
1	0000001 – 1000001	0000001 – 0100001	0000001 – 0010001	0000001 – 0001001	0000001 – 0000101	0000001 – 0000011	0000010 – 0000011
2	...	...	...	...	0000010 – 0000110	0000100 – 0000110	...
3	...	...	...	...	0000011 – 0000111	...	...
4	...	...	...	...	0000100 – 0001100	...	...
...	...	...	...	...	...	...	...
7	...	...	...	0000111–0001111	...	...	...
8	...	...	...	0010000–0011000	...	...	...
...	...	...	...	...	...	...	...
15	...	...	0001111 – 0011111	...	...	...	...
16	...	...	0100000 – 0110000	...	...	...	...
...	...	...	...	...	...	...	...
31	0011111 – 1011111	0011111 – 0111111	0101111 – 0111111	0110111 – 0111111	0111011 – 0111111	0111101 – 0111111	0111110 – 0111111
32	0100000 – 1100000	1000000 – 1100000	1000000 – 1010000	1000000 – 1001000	1000000 – 1000100	1000000 – 1000010	1000000 – 1000001
...	...	...	...	...	...	...	...
63	0111111 – 1111111	1011111 – 1111111	1101111 – 1111111	1110111 – 1111111	1111011 – 1111111	1111101 – 1111111	1111110 – 1111111

Table 7.2: Butterfly input indices written in binary format

	A$_1$	A$_2$	B$_1$	B$_2$	W
1	0, 0,c[4:0]	1, 0, c[4:0]	0, 1, c[4:0]	1, 1, c[4:0]	0, 0, c[4:0]
2	0, 0,c[4:0]	0, 1, c[4:0]	1, 0, c[4:0]	1, 1, c[4:0]	0, c[4:0], 0
3	0, c[4], 0, c[3:0]	0, c[4], 1, c[3:0]	1, c[4], 0, c[3:0]	1, c[4], 1, c[3:0]	0, c[3:0], 00
4	0, c[4:3], 0, c[2:0]	0, c[4:3], 1, c[2:0]	1, c[4:3], 0, c[2:0]	1, c[4:3], 1, c[2:0]	0, c[2:0], 000
5	0, c[4:2], 0, c[1:0]	0, c[4:2], 1, c[1:0]	1, c[4:2], 0, c[1:0]	1, c[4:2], 1, c[1:0]	0, c[1:0], 0000
6	0, c[4:1], 0, c[0]	0, c[4:1], 1, c[0]	1, c[4:1], 0, c[0]	1, c[4:1], 1, c[0]	0, c[0], 00000
7	0, c[4:0], 0	0, c[4:0], 1	1, c[4:0], 0	1, c[4:0], 1	0000000

Table 7.3: Table showing the methodology to obtain the butterfly inputs indices by using a 5-bit counter

7.2   Buffer

The next large unit of the digital baseband processor was the buffer that is used at the interface of the A/D and D/A converters. This buffer is also used for sampling rate conversion. That is the data is pushed onto the buffer at one rate and it is popped off of the buffer at another rate. Specifically during RX the buffer is filled at the fast clock rate and read out at the slow rate. Multiple OFDM symbols are written filled on this buffer for a later off the line processing. As mentioned earlier during the FFT section the fast rate is 100Msps, which was needed for the ADC of the prototype board, and slow rate is 25MHz, which was defined by the prototype board reference clock.

The buffer memory is a dual port RAM with one port dedicated to the fast clk and the other dedicated to the slow clk as shown in Figure 7.9. The fast_slow signal switches the memory between fast and slow operation modes.

The buffer implementation uses read and write pointers. Asserting the read sig-
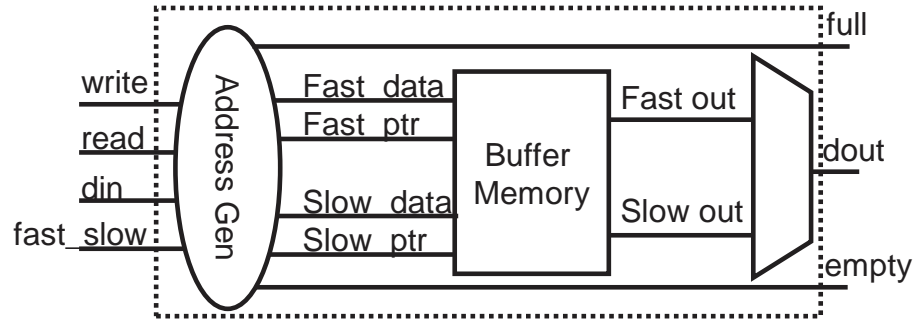
Figure 7.9: Block diagram of the buffer between the baseband digital and baseband analog blocks

nal increments the read pointer and the write signal assertion increments the write pointer. When both pointers are equal the buffer is considered empty. If write_pointer = read_pointer - 1 then the buffer is deemed full and will not allow any more data pushed onto the buffer. Albeit one unfavorable outcome of this scheme is that effectively for a buffer using a RAM with N words, a maximum of N-1 data entries can be pushed onto the buffer.

The read and write pointer values need to be handed over to another clock domain when the buffer is switching from fast operation to slow operation. Even though these domains are synchronized as the fast rate is quadruple of the slow rate synchronizers are needed during this hand over. Cascaded registers are utilized to synchronize the data transferred between clock domains and prevent metastability [63]. Extra latencies due to these additional registers are handled by increasing the transition duration of the buffer by one more cycle.

7.3 Max Selector

Max selector is a very simple block that essentially takes the complex channel tap, computes its absolute value and compares it to a running maximum value and assigns

it as the new running maximum if it is greater. The comparison is performed on the squares of the absolute values. Basic squarers have been used in this architecture for obtaining squares of the real and imaginary parts.

Admittedly a CORDIC block could also have been used to obtain the magnitude of complex CIR data at a lower power and area cost. However multiplier based approach has been used in this block for two reasons.One multiplier based implementation is much much simpler to implement. Two since the application runs at a rather low speed the multiplier is not found to have any restrictive constraints on the system synthesis.

7.4 The Controller

Controller is the finite state machine arbitrating the sequence of events in the ranging system. The events conducted by this block essentially represent the flow of events in Figure 6.1. The VHDL code describing its functionality is included in Appendix. The sequence of events proceed as:

- Buffer filling until the buffer is full.

- Averaging the buffer contents over OFDM symbols until the buffer is empty.

- Computing an FFT by first loading the FFT then starting it and waiting for its completion.

- Once the FFT is done the result are read out and division in frequency domain is executed.

- IFFT is loaded, executed and read out.

- Finally the max selector unit is loaded to find the channel tap with the strongest tap.

The implementation of this state machine is carried out in VHDL. The state machine is represented using a large case statement, which is cased upon the value of a state variable. All outputs are preassigned to default values to prevent latch inference by the synthesizer.

7.4.1 Controller vs. datapath design

One side point that is deemed beneficial to include at this stage is a discussion on design and use of controllers in Digital circuit and system design in low power, low speed digital systems. In most cases the datapath elements in such systems are used very infrequently. Therefore using a high number of data processing elements in parallel would both increase the leakage power and silicon area. Therefore usually shared computational resources are utilized at low power and speed digital design. Implementations of resource sharing often involves use of controllers during loading and unloading the data into and out of the processing elements as well as activating the necessary processing elements and possibly halting the others.

The challenge of such controllers usually lie in their design. This mainly involves correctly identifying latencies and including these in the design. Once the correct timing is achieved in the RTL simulator the synthesis of the controller is usually without any difficulties in its associated static timing analysis (STA). On the other hand the processing elements which are often implemented with a few of lines in RTL code simulates without many difficulties. However during synthesis it is usually such functional blocks that fail static timing analysis (STA) and require redesign of the block by using pipelining or other speed up techniques.

One final difference is with the verification of the controller and datapaths. Datapath components are easier to represent in modeling languages such as Matlab or C such that test vector for verification are easier to generate, whereas for state machines the inherent presence of states complicates modeling and generating test vectors. Moreover representations in other modeling languages offer no real simplifications during their design. Even though there are some facilities to model state machines Simulink/Stateflow and convert them into respective RTL descriptions, such methods do not work for Matlab or C modeling. Therefore these blocks are often best designed using RTL simulators straight from the start.

7.5 Conclusion

The digital baseband sections of the ranging system is fleshed out in this Chapter. The key units of this system are the FFT/IFFT unit as well as memories, buffers and controllers. The FFT unit received significant coverage in this section as the main workhorse of the system. Background information as well as detailed description of its contents are provided. Fixed point details of butterfly and memory units provided with figures. Next the memory sitting at the analog digital boundary is described.The buffer is a straight two pointer first in first out (FIFO) memory implemented from a two port SRAM. Each port is dedicated to operation at one frequency and clock domain crossings are secured by double latching of the domain crossing signals. Max selector takes the absolute value of CIR terms and computes the index of the strongest channel tap. Finally the sequencing of the controller is included as well as some after thoughts on controller vs. datapath design and verification for similar systems.

Chapter 8

Ranging system prototype

The ranging system proposed in the previous section is decided to be prototyped for an on field proof of concept. The prototype is built using Field Programmable Gate Arrays (FPGA) and commercial off the shelf (COTS) analog parts. The prototype is mainly comprised of two boards:

- A baseband board that was originally designed for the Two chip intercom project. [64]

- An RF board that was used by the Multi Carrier Multi Antenna Research project. [65]

Prototype setup is shown in Figure 8.1. In this setup are two transceivers. One transceiver is set to be the transmitter whereas the other is set to be the receiver. This selection is realized by an RF switch connecting the antenna to either the RX or the TX chain.

The RF board has a Phase Locked Loop(PLL) and a Voltage controlled oscillator (VCO) forming the frequency synthesizer outputting a Local Oscillation (LO) at the 2.4-2.5GHz frequency band. On the transmit path of the board, there is an RF
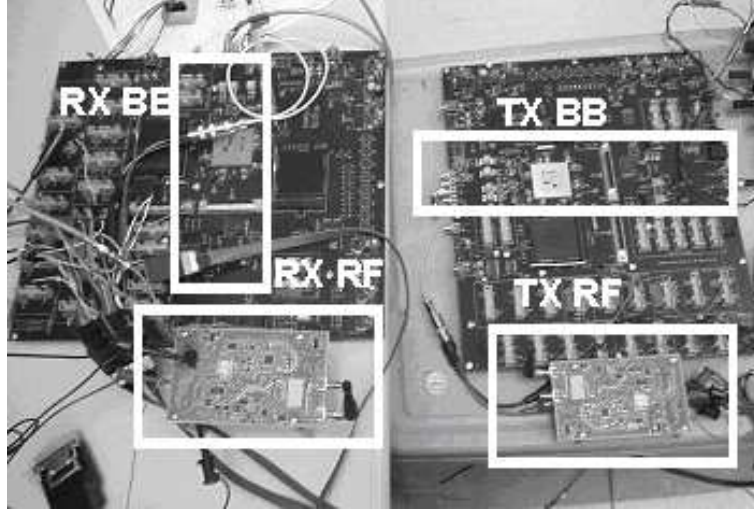
Figure 8.1: Photo of the prototype setup

modulator and a power amplifier (PA). An RF bandpass filter using transmission lines is implemented on board traces. This 80MHz filter determines the band that can be used for the wireless link.As the receive path, it has an RF IC that houses a 2.4GHz Low Noise Amplifier (LNA), mixers and Variable Gain Amplifiers (VGA).

The receiver mixer output filters are on-board third-order low pass discrete LC filters. Originally these low pass filters' cutoff frequency was designed to be 10MHz. However as the ranging application requires a much wider signal bandwidth, the on board inductor and capacitor values are modified such that a 50MHz signal can be down converted using the RF board.

The baseband board hosts an FPGA, an 8-bit, dual-channel, 100Msps Analog to Digital Converter (ADC), an 8-bit, dual-channel, 100Msps Digital to Analog Converter (DAC), ADC preamplifiers and Baluns for differential to single ended conversion of the ADC inputs and DAC outputs. This board additionally hosts the clock generation circuitry that can either use an on board crystal clock or an externally generated clock to generate the 25MHz signal driving the FPGA clock input. The

Function	Used part
FPGA	XCV300E
ADC	AD9288
DAC	AD9709
Preamplifier	AD 8009
VCO	VC9230
PA	RF2126
Receiver	MAX2701
PLL	LMX2326

Table 8.1: Table showing parts used to implement functions

FPGA is a VirtexE300 device that includes on chip DLL clock multipliers by virtue of which 100Msps clocks are generated. The sampling rates of the ADC and DAC limit the maximum achievable signal bandwidth to 50MHz.

Table 8.1 summarizes the important parts that exist on the prototype boards.

8.1   RF and Baseband Board Settings

The RF and Baseband boards have many settings that need to be properly adjusted before measurements could be taken. These settings include: Selecting the frequency control words and programming the PLL through a Microwire serial interface, appropriate transmit or receive mode selections, transmit power selection, LNA gain setting, VGA gain setting, DAC output resistor and voltage selection. The power and gain selections effectively control the prototype received signal strength(RSS) and SNR.

The PLLs are programmed such that the oscillation frequencies are 2.43GHz. In case of nominal frequency codes programmed values the TX and RX Local Oscillator outputs had a 30kHz frequency offset. This frequency offset induces Inter Carrier Interference (ICI) and disrupts the orthogonality of the baseband signal.[66]

To mitigate complications due to frequency offset compensation between the TX and RX oscillator frequencies are programmed to be as close as possible. To this end slightly different frequency codewords are used at each transceiver. The frequency codeword is stepped while observing the frequency offset at baseband using an oscilloscope. Using this method, the lowest frequency offset achieved is close to 3kHz and the ICI and phase rotations due to this offset are negligible.

The control signal in the transmitter is a 3-bit digital bus that sets the power amplifier(PA) output level. This control signal multiplexes the intermediate voltages from a resistive ladder and the multiplexer output drives the PA control voltage. Using this control bus the PA output power can be modified between -21dBm to 0dBm in 3dB steps.

The receive chain also includes signals controlling the gain so that input signals over a wider dynamic range can be handled. First of these control signals is a logic signal that can set the input LNA in a higher or a lower gain setting. The second signal adjusts the gain of the baseband variable gain amplifiers (VGA).

The low and high gain settings of the LNA are at -2dB and 16dB. The VGA gain can be varied from 2dB to 40dB. Much more important than these ranges, these variable gain blocks usually have different noise figures(NF) at different gain settings such that the SNR can be significantly deteriorated at one setting whereas it is slightly modified at other gain settings. As an example, the high LNA gain setting has a NF of 2.3dB whereas the low gain setting has a NF of 16.7dB. Therefore for LNA operation with low LNA gain settings there is a significant SNR degradation.

In experiments using the prototype, the TX output signal was selected at 0dBm to observe a strong signal at the maximum distance of 10m. At this setting, using the high gain(low noise) setting of the LNA caused output clipping. Therefore the LNA was put in the low gain (high noise) mode. The VGA was in a high gain(low
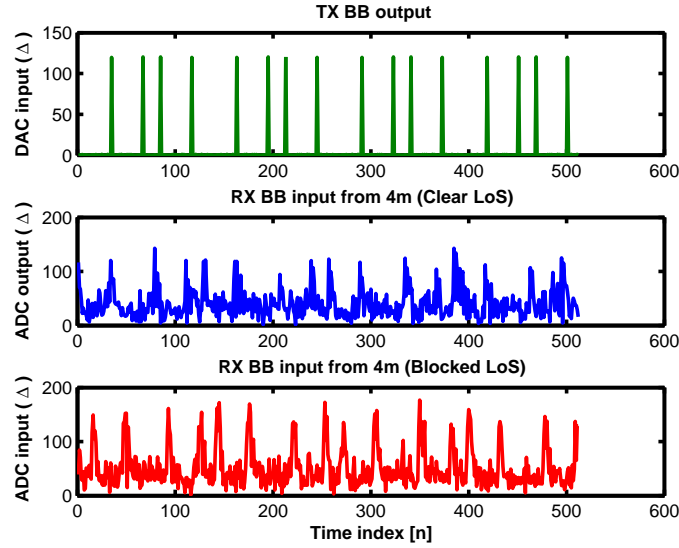
Figure 8.2: Data transmitted and received from 4m, with and without presence of LoS blockers.

noise) mode during the measurements. Due to the low gain and high noise setting of the LNA our prototype incurred a 14dB SNR penalty. It can be argued that to avoid the LNA overloading the TX signal power could be reduced. In this case more than 10dB of TX power reduction is needed to prevent LNA overloading. That is signal power is traded with reduced NF.

Sample transmitted and received signals using this setup are shown in Figure 8.2. Note that, the presence of LoS blockers such as dry walls or wooden doors does not significantly attenuate the signal. Four periods (or OFDM symbols) of data are acquired per sampling window. This is equivalent to a 5.12ms sampling time window. The SNR computed from these data for the clear LoS case is -3dB. One reason for this SNR is the high noise figure due to the low gain setting of the LNA.

Before analyzing the prototype data, there are some more practical issues that remain to be addressed.
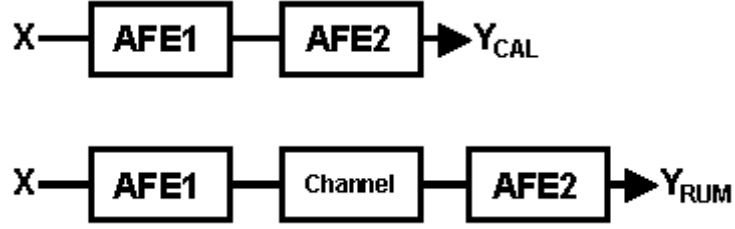
Figure 8.3: Acquisition of CAL and RUM data sequences

8.2 Calibration

Before data analysis, the system must be calibrated to remove the delay introduced by the analog front ends (AFE) of the TX and RX. This delay is caused by channel selection filters, parasitic filters due to board traces, RF amplifiers, antenna responses, etc. To cancel AFE effects, two sets of measurement data are needed. The first set is called the *calibration* (CAL) data and is acquired when the transmitter and receiver are right connected to each other (Figure 8.3). The connection can be a short ($l < 30cm$) coaxial cable or can even be wireless. This calibration data and/or its relevant parameters are stored. The second set is the data received from the transmitter, which is now at the distance to be measured. This set is called the *range under measurement* (RUM) data. The main difference between CAL and RUM data sets is that CAL data includes only the effects of the analog front ends at the transmitter and receiver, whereas the RUM data also includes the wireless channel effects Figure 8.3.

8.3 Clock Offsets

Due to physical limitations in the setup, one-way measurements are taken instead two-way time transfers. Then, an oscilloscope is used to measure the clock offset

between the two nodes. The offset is defined as the delay between symbol edges, which mark the beginning of a TX or RX period. The offset is then used to correct the ToF measurement. The offset changes during the time it takes to capture data and then to measure the offset. Thus, to stabilize the clock offset, a single clock is fed into both nodes so the offset remains constant throughout each experiment. In real operation, however, the forward and reverse measurements would take place within a fraction of a second and the frequency offset can be considered constant for that interval.

8.4   Interference in Prototype

Since the prototype operates in the crowded 2.4GHz ISM (Industrial Scientific and Medical) band, there is interference from other devices using this band. The possible interferers include 802.11b/g network elements, Bluetooth radios, Microwave ovens etc...

Particularly strong interference is from an 802.11b/g WLAN access point located right above the setup. When turned on and idle there are regular bursts coming from this access point (around every 20ms). These are the channel monitoring bursts used to regulate media access functions. In this mode the idle time of the access point long enough to acquire the entire 5.12 μs sampling window. What is more this 20ms window is even long enough to realize a two way communication which is expected to last around 50μs. However if the 802.11b/g access point is active or idle but monitoring the channel during data acquisition then the acquired data is practically useless. This situation is illustrated in Figure 8.4. Here the received signal is completely saturated when the access point is also transmitting. In this case the simplest solution is detecting the presence of the interferer, due to the high
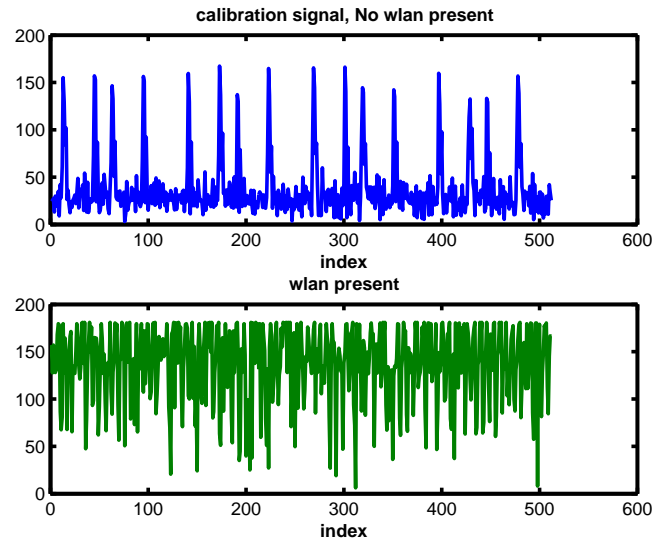
Figure 8.4: Effect of the WLAN interference on the received signal

energy of the received signal, and simply discarding the acquired data set. The measurement should be repeated until there are no interference. However as the WLAN gets more active, more range measurements are unusable. Therefore avoiding operation at locations of heavy WLAN activity such as close proximity of active access points may be wise for the current ranging system.

One important point to keep in mind is that the WLAN access point is fairly high powered and is located right above the measurement setup. Had the access point been situated further from the setup the destructive effects of the access point transmission could be lower saturate the received ranging signal.

Aside from the WLAN network there are not any other significant interference sources detected while prototyping. Among the possible interferers obviously a microwave oven would have catastrophic effects on the ranging system by blasting power in the frequency of operation. In case of a bluetooth interferer the interferers lower power than WLAN access points and they may not be as easy to detect at the front

end. Blutooth signals are GMSK pulses with 1MHz bandwidth and they switch frequency every 625μs [67]. Therefore a bluetooth interferer pair will likely corrupt only two to four subcarriers during the data acquisition and are not expected to have catastrophic effects on the scheme aside from performance degradation. However additional bluetooth transceivers will definitely degrade the performance to even lower levels.

One final interference concern is the self interference from neighboring nodes. If immediate neighbors are trying to range distances to their neighbors simultaneously there can be interference from neighboring sensor nodes. Since the nodes knows of their neighbors and an arbitration mechanism is already in place for the media access in the data communication radio such information can be shared with the ranging system when arbitrating the ranging measurements in the network.

8.5   Prototype Data Analysis and Results

Prototype measurements were taken as the distance between TX and RX was swept from 1.5 to 10m. To obtain ranges from this data, CAL and RUM datasets need to combined to remove AFE effects. This can be performed in two ways:

8.5.1   Remove AFE frequency response

In the first method, the CAL data is treated as the input waveform and it is removed from the RUM data. To explain this, consider the transfer functions for

Figure 8.3.

$$Y_{CAL}(\omega) = H_{AFE1}(\omega) \times H_{AFE2}(\omega) \times X(\omega) \tag{8.1}$$

$$Y_{RUM}(\omega) = H_{AFE1}(\omega) \times H_{channel}(\omega) \times H_{AFE2}(\omega) \times X(\omega) = H_{channel}(\omega) \times Y_{CAL}(\omega) \tag{8.2}$$

Viewing the channel as a linear system, the CAL data can be considered as an input that is filtered by the channel to make the RUM output. Thus, dividing the RUM data by the CAL data yields the frequency response of the wireless channel and conversion back to the time domain yields the CIR. Figure 8.5 shows the estimated CIR for a 4m and a 6.5m test. Note, as can be seen in the 6.5m plot of Figure 8.5, the strongest CIR tap is not always clearly dominant. If there is a clearly dominant channel tap then the ToF can be computed as:

$$ToF = n_{st} - [n_{cur\_RUM} - n_{cur\_CAL}] - [OS_{RUM} - OS_{CAL}](mod128) \tag{8.3}$$

where, n_{st} is the index of the strongest tap in CIR; n_{cur_RUM} , n_{cur_CAL} are the indices of the OFDM symbol edges for RUM,CAL data sets and OS_{RUM} , OS_{CAL} are the TX-RX clock offsets measured with an oscilloscope for RUM, CAL data sets.

8.5.2   Remove AFE delay

In the second method, the CIRs corresponding to both CAL and RUM data sets are determined separately. From these responses, the strongest channel taps are selected as the LoS taps. Figure 8.6 shows the estimated CIRs for CAL data and RUM data acquired when the range is 4m. Here the strongest taps are clearly dominant. Then the time of flight delay can be calculated as:

$$ToF = [n_{st\_RUM} - n_{st\_CAL}] - [n_{cur\_RUM} - n_{cur\_CAL}] - [OS_{RUM} - OS_{CAL}](mod128) \tag{8.4}$$
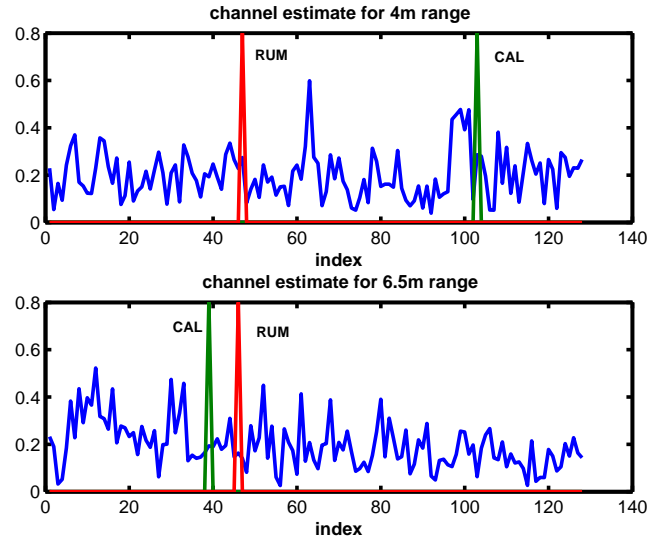
Figure 8.5: Estimated channel using the first method. Shown are estimated channels for 4m and 6.5m ranges respectively. Labeled spikes represent the symbol edges for the CAL and RUM data

where, n_{st_RUM}, n_{st_CAL} are the indices of strongest RUM, CAL taps and n_{cur_RUM}, n_{cur_CAL} , OS_{RUM} , OS_{CAL} are the same as Section 8.5.1.

In summary, the first approach tries to remove the AFEs altogether and estimate the whole channel. In contrast, the second approach just tries to remove the extra delay due to the AFEs. In a way, the first method tries to obtain more than is necessary, while the second method extracts only the needed data.

Also notable is the reduced need for data storage while removing the CAL delay. In the first method, storage of the whole calibration sequence is required and it is used each time ToF to be calculated. However in the second method, only the delay calculated from the calibration sequence needs to be stored. Additionally, the second method has a simpler implementation if the pilot spectrum is selected such that the division operations can be reduced to simple arithmetic shifts.
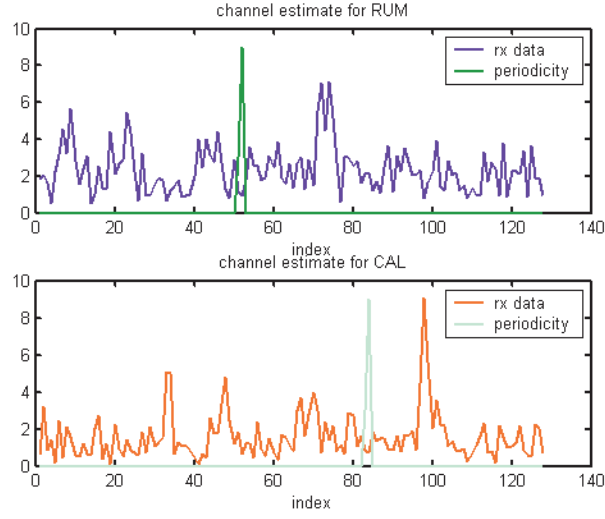
Figure 8.6: Estimation of RUM and CAL channels separately at range of 4m. Also shown are the symbol edges in each data window.
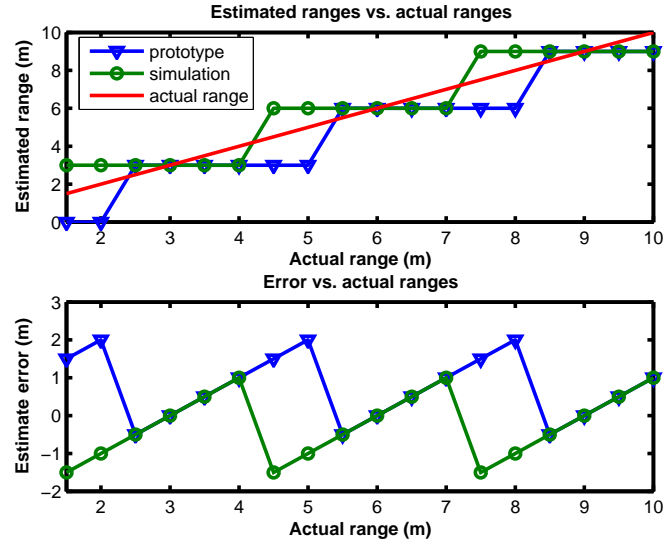

Figure 8.7: Distance measurement results and corresponding errors. Also shown are the range estimates and associated errors from the simulations of the ranging system

8.6 Prototype Results

Using the *"Remove AFE delay"* analysis method, the ranges were estimated for the TX-RX distances between 1.5m and 10m. The results are plotted in Figure 8.7 and they indicate that the prototype can detect ranges in 3m increments. This result is in agreement with the observation that the sampling period of 10ns corresponds to a 3m distance. There were many the nonidealities that were not included in the simulations as well as nonidealities whose magnitudes could not be estimated properly. Nevertheless the functionality of the prototype proves none of those effects were deal breakers and the ranging system is robust against the existing nonidealities.

This result also is in agreement with the simulated results shown in the same figure. The error for the prototype measurements is within the range of [-2m, 0.5m]. On the other hand the range estimate errors in case of simulations were in the range of [-1.5m, 1m]. Therefore the bias of the estimate errors is different for the prototype and the simulations.

That is, the prototype measurements display a rounding down (flooring) quantization behavior, whereas the simulation results display a rounding quantization behavior. The discrepancy is attributed to the modeling in the simulations. The simulated channels were created as oversampled models and resampled to the ADC sampling rate of 100Msps. This used `resample` function causes the rounding quantization seen in the simulations.

To put these results in perspective the results from the prototype are plotted on a graph together with other ranging systems. This graph is included in Figure 8.8. Here the horizontal axis represents the bandwidth of the system in MHz and the vertical axis represents the resolution of the ranging system.Both axes are drawn on a logarithmic scale. The results of this work is marked on the graph by an arrow. The downward sloped line is the CRLB of the range measurement calculated over the
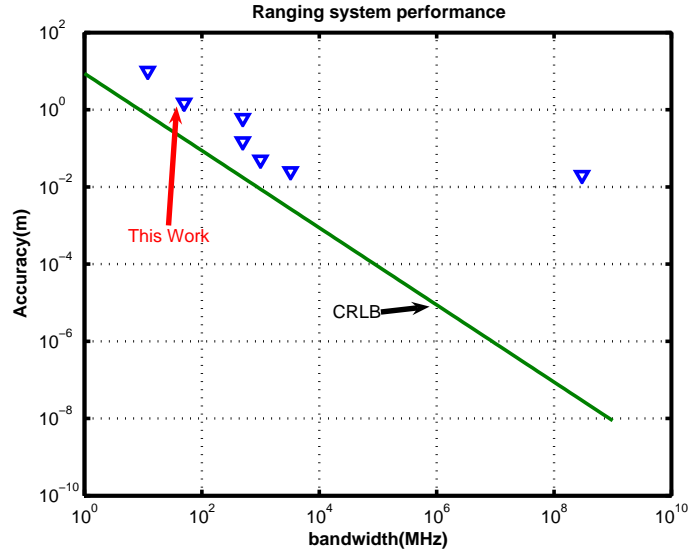
Figure 8.8:   Comparison of this ranging system against ranging systems from literature.

frequency band at a 7dB SNR. To elaborate on the other particular points on the graph. The upper left point is the GPS performance with its pseudorange measurement range error [1]. The cluster of four data points are taken from UWB systems described in literature [55, 68, 69, 70]. The leftmost entry is the Wimedia alliance ranging accuracy performance whereas the other three to its right are pulse based UWB systems. The outlier to the very right is a laser range finder application. [71]. This system lies quite far from its theoretical bound defined by the CRLB of the range and thus is quite inefficient for ranging purposes. However it can still offer a precise measurement due to the high speed signals that are used.

8.7   ASIC Cost Estimates

Once the functionality of the ranging system was verified by the measurements taken from the ranging system prototype A question that needs to be answered is the

power consumption of such a system. The target power consumption was specified in Chapter 1 as a 35-40mW of maximum power dissipation. Therefore in the real application it is desirable to keep the power of the ASIC under such a number.

The obvious answer of determining the power of the designed ranging system can be looking at the power consumption of the prototype. However this would be a quite different quantity than the power consumption for an integrated solution. There so many reasons regarding the improvements that can be accomplished with an integrated solution is that it almost does not need any justification.

The main reason for the discrepancy between prototype power dissipation and an integrated solution is that each of the COTS parts used in the two prototype boards were generic parts that were to be designed by a spectrum of applications. Therefore to accommodate this range of applications these chips were designed with flexibility and flexibility generally leads to waste of power. The next problem was that there were many voltage regulators on these boards since these COTS parts were designed to operate on different voltages. Voltage regulators usually operate inefficiently and could be avoided or reduced to a single regulator in case of an integrated implementation. Therefore as the last part of the implementation of the ranging unit sample power dissipations are estimated for an integrated implementation. These estimates were considered in two cases as estimates for the Digital baseband and the Analog/RF sections.

8.7.1   Digital baseband Estimates

The first part of the power estimation efforts the digital baseband power consumption is obtained. To this end the RTL descriptions of the digital baseband blocks that were used to program the FPGA are resynthesized using a 90nm ASIC library. The design is synthesized in Design Compile®. Once the gate level netlist is obtained

	Power	Area	Nominal Leakage
Synthesized memories	5mW	0.9 mm^2	0.13mW
Memory macros	2.35mW	0.25 mm^2	0.64mW

Table 8.2: Digital power consumption estimate

signal activity factors are calculated on this netlist with RTL simulations. Once the switching activities are calculated and back annotated the total power consumption estimates are computed using Power Compiler ®.

It should be noted that the clocks were gated such that inactive sub blocks did not toggle the clock inputs of the registers within. To realize this functionality clock gate cells from the standard cell library were utilized. The clocking rate was again 25MHz for the ASIC synthesis.

The synthesis and power estimation was done for two cases. In the first case all the storage functions in the digital baseband is realized by using registers. That is the buffer memory as well as the FFT memories all were implemented as register files. For the second case the FFT memories buffer memory as well as the other memories are implemented in memory macro modules received from the ASIC library vendor.

Table 8.2 shows the result of the synthesis for these two cases. The power consumption and silicon area estimates are lower for the case utilizing memory macros. This is an intuitive result since the memory macro would be a dense implementation it is expected to occupy a smaller area. Also the memory has measures to minimize clocking load on its contents whereas in the register file implementation an turned on unit will present all its registers on the clock tree. Therefore lower power memory implementations are expected as well.

One unexpected result was the higher leakage power dissipated with the implementation using memories. Even though memories were expected to be less leaky since they were also optimized for high speed operation even though not really needed in

this application. however these high optimizations inadvertently increase leakage.

As seen from these numbers the digital baseband section dissipates very little active power dissipation as well leakage, which can even be eliminated by leakage reduction techniques. Therefore its power consumption is very little with comparison to the total budget. Hence the critical unit for power consumption is the analog and RF sections of the ranging unit.

8.7.2   RF/Analog power estimates

As the second part of the ranging system the power consumption of its analog and RF sections need to be estimated. The key observation regarding the RF section of this unit is that it operates at 2.4GHz frequency band with baseband bandwidth of 50MHz. This is similar to the Bluetooth system which has a 80MHz bandwidth at the 2.4GHz band [67]. Therefore the RF front end specifications of the two systems are comparable and the Bluetooth RF front number power numbers can be useful as benchmarks for the RF unit required by the ranging system.

Table 8.3 presents a little survey of RF front ends for Bluetooth applications that has been published over the last years. According to this table there has been RF front ends around 30mW - 40mW of power consumption. It is worth to note that usually published Bluetooth chips include the baseband GMSK filters as well as sophisticated frequency synthesizers as Bluetooth is a frequency hopping spread spectrum system. However in the ranging system there are no Gaussian pulses or different RF frequencies involved. Therefore a possible ranging front end would be devoid of these complicated frequency synthesizers as well as the baseband demodulators. On the other hand 1MHz of Bluetooth baseband bandwidth is much less than the 50MHz bandwidth used in the ranging system. Hence in the ranging system wider bandwidth requires baseband amplifiers possibly with higher power consump-

	Total Power	RF front end	VCO	PLL	Tech
Filiol [72]	34mW	8mW	6mW	3.5mW	SiGe
Cojocaru [73]	43mW	11.3mW		14.6mW	BiCMOS
Zolfeghari [74]	34mW	10.5mW		13mW	0.25μ CMOS

Table 8.3: Table showing the power dissipation of RF front ends with similar specifications to the ranging system.

tion. Overall the takeaway from Table 8.3 is that it is possible to design an RF front end for the proposed systemfor 30mW of power dissipation.

In addition to the RF sections power consumption estimates due to the analog to digital conversion needs to be included in the overall analog front end power. To estimate this quantity a figure of merit metric often used to compare A/D converters is utilized. This metric often denoted as FOM is defined as

$$\frac{P}{f \times 2^{ENOB}} \tag{8.5}$$

where P is the A/D converter power consumption, f is the conversion rate and ENOB is the effective number of bits i.e the resolution of the converter [75]. This FOM metric is usually specified in pJ/conversion and it is a measure of energy spent per Analog to Digital conversion. For fast and low resolution A/D converters typical FOM are around 1 pJ/conv [69], while best designs achieve as low as 0.2pJ/conv [69]. Therefore by assuming a moderate 0.5pJ/conv FOM and 130Msps of sampling rate and 6 bits of A/D resolution in Equation 8.5 a power estimate of 4.3mW is obtained. Considering I and Q channels of the receiver the total A/D conversion power estimate becomes 8.6mW.

Since the Analog frontend would be on only during the data acquisition phase followed by an offline computation in the digital unit the power consumption can be broken into two phases in the first of which only the analog power is dissipated and the second of which is only digital power consumption. Hence during the acquisition phase only the Analog front end is active the power dissipation estimate is 38.6mW

	During Acquisition	During computation
	30mW(P_{RF})	2.35mW(P_{DIG})
	8.6mW(P_{ADC})	
P_{TOTAL}	38.6mW	2.35mW

Table 8.4: Power consumption of the overall ranging system in Acquisition and computationmodes

whereas during the digital computation phase the power estimate is 2.35mW. These results are summarized in Table 8.4, where P_{RF} is the RF block power consumption. P_{ADC} is the ADC converter power dissipation, P_{DIG} is the digital baseband power and P_{TOTAL} is the total power dissipation. Finally as conclusion it can be stated that this ranging system is estimated to be consuming a power within the power budget of less than 40mW stated earlier.

8.8 Conclusion

In this Chapter implementation aspects of the ranging system was explored. The system was prototyped on a platform platform built from COTS parts. In the beginning of the chapter this prototyping platform was described in detail. Next presented were methods that were used in this prototype to overcome the clock offsets and calibration methods to remove analog front end effects. The calibration was realized by removing the delay due to analog front end filtering effects. Next discussion included the acquired data and its analysis. Following was a presentation of the results where the functionality of the ranging system was proven and its measured performance was compared against a number of different ranging systems. Finally before concluding the chapter an section estimating the power consumption in case the prototyped system was integrated on silicon. The result of these estimates revealed that the analog front end of the ranging system can be expected consume a power of 38mW whereas

the digital baseband sections are expected to consume 2.5mW of power. Hence the system can stay within the power budget specified in earlier chapters.

Chapter 9

Future Work and Conclusion

In this thesis implementation of localization system for sensor networks have been presented. To begin with, it is identified that given a set of reference points localization consists of two sub tasks: Establishing relationships to the reference points and algorithmically computing the unknown position using the reference points and their relations to the unknown position.

The first step taken was going through a set of localization algorithms designed for sensor network localization. The algorithms are divided into two classes centralized and distributed localization algorithms. After a detailed review of different algorithms a distributed two phase localization algorithm was selected for implementation. This algorithm combined advantages of ranging based methods as well as connectivity based methods to achieve accuracy and robustness of the algorithm.

Next design of the HopTerrain based localization system is presented. This algorithm is the first phase of the localization system that was selected for implementation. It performs triangulation using number of hops from reference points. Additionally it can share its resources with the refinement phase of the algorithm. This is because it can be easily adapted to perform triangulations with Euclidean distances.

The chapter started with a mathematical formulation of the triangulation problem. Following this, methods on its solution are considered and QR decomposition based LS solution is selected for realization. QR decomposition is realized with Givens rotations. Following the LS solution is selected functional additions are included in the unit that will allow computation of the Hop Distances are discussed. These additions include units that can create network packets to be used in hop count determinations.

The detailed implementation of the proposed Hop count based localization system is presented next. Various alternative implementations were considered. The design subunits were defined in detail. The final design exhibits 1.7mW of active power dissipation. This implies an order of magnitude active power dissipation savings with respect to a General-purpose microprocessor or DSP implementation. This shows the low power dissipation goal of the implementation has been achieved. In addition the system occupies 0.79mm^2 of silicon area and the fixed-point implementation causes a negligible degradation in the accuracy of the final location outputs.

Once the triangulation system implementation is complete the second part of the localization problem, ranging, is taken up. In this second half of the thesis ranging for sensor network localization has been systematically studied. The work starts with a classification of ranging methods in terms of how they view the wireless channel. Here ranging methods using the timing view of the channel are found to be appropriate for their robustness against multipath effects. Next shortcomings with this view are addressed. Fundamental performance limits are investigated, OFDM based signaling is selected for its low computational complexity and two way time transfer was employed for TX/RX clock synchronization.

Once the algorithm to be implemented was decided the next step becomes selecting the relevant parameters of the algorithm. In this step these parameters of the ranging

system are decided. The implemented system uses an OFDM based 128-pt multi carrier system that is sampled at 100Msps. The signal bandwidth is 50MHz and receiver ADC has 6bits of resolution. The signal fits and operates in the 2.4 GHz ISM band. This proposed ranging system has been simulated in Matlab for performance evaluations. As a result the algorithm is found to be functional with ranging error in the range of $[-1.5\text{m}, 1\text{m}]$. Now that the system is designed and all parameters are selected. The next step is to realize its digital baseband.

The key units of this ranging system are its FFT/IFFT unit as well as memories, buffers and controllers. The FFT unit received significant coverage in this section as the main workhorse of the system. Background information as well as detailed description of its contents are provided. Fixed point details of butterfly and memory units provided with figures. Next the memory sitting at the analog digital boundary is described.The buffer is a straight two pointer first in first out (FIFO) memory implemented from a two port SRAM. Each port is dedicated to operation at one frequency and clock domain crossings are secured by double latching of the domain crossing signals. Max selector takes the absolute value of CIR terms and computes the index of the strongest channel tap. Finally the sequencing of the controller is included as well as some after thoughts on controller vs. datapath design and verification for similar systems.

In the final parts of the thesis implementation aspects of the ranging system was explored. The system was prototyped on a platform platform built from COTS parts. In the beginning of the chapter this prototyping platform was described in detail. Next presented were methods that were used in this prototype to overcome the clock offsets and calibration methods to remove analog front end effects. The calibration was realized by removing the delay due to analog front end filtering effects. Next discussion included the acquired data and its analysis. Following was a presen-

tation of the results where the functionality of the ranging system was proven and its measured performance was compared against a number of different ranging systems. Finally before concluding the dissertation a section estimated the power consumption in case the prototyped system was integrated on silicon. The result of these estimates revealed that the analog front end of the ranging system can be expected consume a power of 38mW whereas the digital baseband sections are expected to consume 2.5mW of power. Hence the system can stay within the power budget specified in earlier chapters.

To conclude in this thesis the two critical tasks of localization was realized. For the first part a 1.7mW triangulation computation implemented that achieves much more power efficiency than any programmable device. The second critical task of implementing the ranging system was accomplished by using pure radio signals that is without need for any acoustic or ultrasonic equipment. In addition this system achieves a [-2m, 0.5m] ranging accuracy which can be converted to an acceptable position estimate with adequate accuracy by the triangulation system. Finally the power consumption estimate of the ranging system in the worst case is 38mW.

9.1   Future Work

As with many engineering endeavors it is hard to conclude the completion of a given task. Even though there are initial specifications which are eventually met there is always room for possible improvements on the design by squeezing out the last mW of power of improving the performance by 10% more. The endless chase of more performance at lower costs have been at the cause of all these efforts. In this vein, there are possible extensions and future work which would be helpful improving the results of this thesis. This possible future work can be grouped in three groups.Improvements

at the algorithmic or system level, on the prototype and improvements that can be achieved with an integrated implementation.

9.1.1   System level improvements

The most obvious system improvement follows the observation that the design in Section 6.2 set signal bandwidth at half the sampling rate effectively forcing Nyquist rate sampling. However the signal bandwidth determines CRLB whereas the sampling rate determines its own error. That is the signal bandwidth can be selected to satisfy the CRLB then the sampling rate can be increased to achieve delay quantization error be within limits or close to the CRLB itself. This may require an oversampled signal that is sampled at a higher rate than its Nyquist rate. The advantage of such an approach can be signals with narrower bandwidths can be employed and transmission at lower frequencies can be possible. This would both simplify the RF circuit design as well as baseband analog circuits and allow power reductions. However the A/D conversion still needs to be aggressive. The advantage of this scheme is the decoupling of sampling rate and signal bandwidth.

Another way of looking at this phenomenon is the signal is being looked at a finer time resolution than its Nyquist rate. An additional way of achieving such a result is interpolating a signal sampled at the Nyquist rate. The penalty incurred for such an approach would be the computational complexity due to the required interpolation filters [43]. Furthermore combining oversampling and interpolation can improve performance even further at the expense of high sampling and additional computational complexity.

9.1.2 Ranging prototype improvements

The second group of possible improvements involve changes that can be made on the prototype. The simplest improvement involves efforts to get more SNR from the wireless link. It was stated during the prototype description that LNA needed to be kept in the low gain mode due to saturation of its output. One possible remedy is stepping down the transmit power until the LNA output is not clipped in the high gain setting. If the transmit power at this point is more than -13dBm, which is the Noise figure difference between high and low gain setting of the LNA. Than the overall link SNR improves. However if the transmit power is less than -13dBm than the gain in reduced noise is eaten away by the reduction in signal power.

Second possible improvement also involves gain and controlling gain. Adding an AGC loop can be useful in setting the programmable gains in the system automatically. It would also reduce the noise figures such that there would not be any SNR penalty for operating in a low gain setting when not needed. Since such a loop would necessitate amplitude and peak detectors these blocks can also be used to detect presence of interferers since they may add significant noise content onto the received signal. Once interferers are detected the measurements can be discarded to be repeated later.

A third prototype improvement is possible if a secondary communication link is established such that data can be exchanged between the transmitter and the receiver. In such a case real two way ranging measurements can be executed. This is because the forward and reverse communication needs to be executed in rapid succession which is much faster than any triggering that can be achieved by humans.Therefore an electronic sequencing is needed to adjust the order of these transmissions. The secondary data exchange radio is needed for changing the time of flight measurements but more importantly to exchange handshaking data that allows requesting acquisi-

tions and acknowledging their completions. This communication mechanism can be a secondary radio as would be in the case of a fully operational sensor node. However for prototyping purposes in can be cables connecting the transmitter and the receiver such that simple communications can be realized.

9.1.3 Realization improvements

One step that could bring a lot of improvements is the complete integrated implementation of the localization system along with its digital, analog and RF sections on a single chip. Digital based band sections of the ranging system was designed in dedicated logic and could be realized on an ASIC without difficulty. However the custom designing analog and RF sections in addition to the dedicated digital design, proves challenging for a single designer and needs much more designer hours. If such an integrated system can be designed its bandwidth, sampling rate are the first parameters that can be optimized. In such a custom design case, higher bandwidths, sampling rates and accuracies can be accommodated while achieving lower power levels. Also again in such an implementation the triangulation unit can be integrated with the digital baseband of the ranging system to provide complete localization functionality. Therefore a fully integrated implementation is the next logical step for improved ranging and full localization capability.

The last possible future work can involve using the UWB radios that are already designed for the MultiBand OFDM flavor of UWB systems [55]. These systems are already wideband and they already utilize multi carrier signalling. The parts already available for these applications can be programmed. Therefore it is possible to leverage these parts for ranging purposes with means of mere software modifications.

Bibliography

[1] Elliot Kaplan, editor. *Understanding GPS Principles and Applications.* Artech House Publishers, Norwood, Massachusetts, 1996.

[2] James J. Caffery Jr. *Wireless Location in CDMA Cellular Radio Systems.* Kluwer Academic publishers, Boston, Massachusetts, 2000.

[3] Jeffrey Hightower and G. Boriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.

[4] L. Doherty, K. Pister, and L. El Ghaou. Convex position estimation in wireless sensor networks. In *IEEE Infocom*, Anchorage, AK, April 2001.

[5] X. Nguyen, M. Jordan, and B. Sinopoli. A kernel-based learning approach to ad hoc sensor network localization. *ACM Transactions on Sensor Networks*, 1(1):134–152, 2005.

[6] J. Rabaey et al. Picoradio supports ad-hoc ultra-power wireless networking. *IEEE Computer*, 33(7), July 2000.

[7] K. Langendoen and N. Reijrs. Distributed localization in wireless sensor networks: A quantitative comparison. *Computer Networks (Elsevier), special issue on Wireless Sensor Networks*, November 2003.

[8] J. Van Greunen and J. Rabaey. *Locationing and Timing Synchronization Services in Ambient Intelligence Networks,in Ambient Intelligence: W. Weber, J. Rabaey, and E. Aarts, Ed.* Springer-Verlag, 2005.

[9] A. Savvides, L. Girod, M. Srivastava, and D. Estrin. *Localization in Sensor Networks, a chapter in the book Wireless Sensor Networks, Edited by Znati, Radhavendra and Sivalingam.*

[10] R. Christ and R. Lavigne. Radio frequency based personnel location systems. In *International Carnahan conference on Security Technology*, pages 141–150, 2000.

[11] S. N. Simic and S. Sastry. Distributed localization in wireless ad hoc networks. *UC Berkeley ERL report*, 2001.

[12] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine, Special Issue on Netowrking the physical world*, August 2000.

[13] N. Bulusu, J. Heidemann, V. Bychkovskiy, and D. Estrin. Density-adaptive beacon placement algorithms for localization in ad-hoc wireless networks. In *IEEE Info-com*, New York, NY, June 2002.

[14] J. Beutel. Geolocation in a picoradio environment. Master's thesis, UC Berkeley, December 1999.

[15] C. Savarese. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. Master's thesis, UC Berkeley, May 2002.

[16] D. Niculescu and B. Nath. Ad-hoc positioning system. In *Globecom Conf*, November 2001.

[17] C. Savarese, J. Rabaey, and J. Beutel. Locationing in distributed ad-hoc wireless sensor networks. In *Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, pages 2037 – 2040, Salt Lake City, UT.

[18] J. McLurkin. Algorithms for distributed sensor networks. Master's thesis, UC Berkeley, May 1999.

[19] C. Savarese, K. Langendoen, and J. Rabaey. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. Monterey, CA, 2002.

[20] S. Capkun, M. Hamdi, and J.P. Hubaux. Gps free positioning in ad hoc networks. In *Hawaii Internationa Conference on System Sciences*, May 2001.

[21] A. Savvides, H. Park, and M. Srivastava. The Bits and Flops of the N-hop Multilateration Primitive For Node Localization Problems. In *First ACM Inter. Workshop on Sensor. Networks and App.*, Atlanta, September 2002.

[22] J. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.

[23] G. Golub and C. Van Loan. *Matrix COmputations*. Johns Hopkins, 1996.

[24] A. Bjorck. *Numerical Methods For Linear Least Squares Problems*. 1996.

[25] J. E. Volder. The cordic trigonometric computing technique. *IRE Trans. Electron Comput*, EC-8:330–334, 1959.

[26] Information technology open systems interconnection basic reference model. Technical Report ISO/IEC 7498-1:1994, 1994.

[27] M. Sheets, F. Burghardt, T. Karalar, J. Ammer, Y.H. Chee, and J. Rabaey. A power-managed protocol processor for wireless sensor networks. In *IEEE VLSI systems conference*, Honolulu, HI, June 2006.

[28] E. A. Swartzlander. *Computer Arithmetic.* IEEE Computer Society Press, 1990.

[29] K. Camera. Sf2vhd: A stateflow to vhdl translator. Master's thesis, UC Berkeley, May 2001.

[30] M. Sheets. *Standby Power Management Architecture for Deep-Submicron Systems.* PhD thesis, UC Berkeley, May 2006.

[31] K. Kuusilinna, C. Chang, J. Ammer, B. Richards, and R. W. Brodersen. Designing bee: a hardware emulation engine for signal processing in low-power wireless applications. *EURASIP Journal on Applied Signal Processing, special issue on Rapid Prototyping of DSP Systems*, 2003.

[32] L. T. Clark et.al. An embedded 32-b microprocessor core for low-power and high-performance applications. *IEEE Journal of Solid-State Circuits*, 36(11):1599 – 1608, November 2001.

[33] Texas Instruments, MSP430F11x1A Mixed signal Microcontroller datasheet, Literature No: SLAS2416.

[34] J. Moon et. al. An area-efficient standard-cell floating-point unit design for a processing-in-memory system. In *European Solid-State Circuits Conference*, September 2003.

[35] J. Draper et. al. Implementation of a 256-bit wide word processor for the data intensive architecture (diva) processing-in-memory(pim) chip.

[36] Texas Instruments TMS320C6713- TMS320C6713B Floating point Digital Signal Processors datasheet. Literature No: SPRS186H.

[37] Texas Instruments TMS320C6713/12C/11C Power Consumption Summary, Literature No: SPRA889.

[38] C.M. Rader. Vlsi systolic arrays for adaptive nulling. *IEEE Signal Processing Magazine*, 13(4):29 – 49, July 1996.

[39] H. Leung and S. Haykin. Stability of recursive qrd-ls algorithms using finite-precision systolic array implementation. *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, 37(5), May 1989.

[40] T. S. Rappaport. *Wireless Communications: Principles and Practice,2 nd ed.* Pearson Education International, 2002.

[41] J. G. Proakis. *Digital communications, 4th ed.* McGraw and Hill, 2001.

[42] J. Barry, D. Messerschmitt, and E. Lee. *Digital Communications, 3rd ed.* Kluwer Academic Publishers, 2003.

[43] H. Meyr, M. Moenclaey, and S. Fechtel. *Digital Communication Receivers.* John Wiley and Sons, 1998.

[44] H. Van Trees. *Estimation Detection and Modulation Theory, Vol. 3.* Wiley, New York, 2001.

[45] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *ACM MobiCom*, Boston, MA, 2000.

[46] I. Maravic, J. Kusuma, and M. Vetterli. Low-Sampling Rae UWB Channel Characterization and Synchronization. *Journal of Communications and Networks*, 5(4):319–327, December 2003.

[47] S. Gezici, Z. Tian, G.B. Giannakis, H. Kobayashi, A.F. Molisch, H.V. Poor, and Z. Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *Signal Processing Magazine, IEEE*, 22(4):70 – 84, July 2005.

[48] K. Whitehouse, C. Karlof, A. Wooand F. Jiang, and D. Culler. The effects of ranging noise on multihop localization: an empirical study. In *International Conference on Information Processing in Sensor Networks (IPSN '05)*, Los Angeles, CA, April 2005.

[49] Chipcon, CC2431 System on chip for 2.4GHz ZigBee/ IEEE802.15.4 with location engine datasheet, Texas Instruments Literature No: SWR034A.

[50] T. Kimura, S. Wadaka, K. Misu, T. Nagatsukaa, T. Tajime, and M. Koike. A high resolution ultrasonic range measurement method using double frequencies and phase detection. In *IEEE Ultrasonics Symposium Proceedings*, 1995.

[51] T. Karalar. An acoustic digital anemometer application. Master's thesis, UC Berkeley, May 2002.

[52] M. Maroti, B. Kusy, G. Balogh, P. Volgyesi, K. Molnar, A. Nadas, S. Dora, and A. Ledeczi. Radio interferometric positioning. In *ACM Third International Conference on Embedded Networked Sensor Systems (SenSys 05)*, pages 1–12, San Diego, CA, November 2005.

[53] FCC, First Report and Order, FCC 02-48, February 2002.

[54] R. Van Nee and R. Prasad. *OFDM for Wireless Multimedia Communications*. Artech House Publishers, 1998.

[55] MB-OFDM Proposal Specifications for IEEE MultiBand OFDM Physical Layer Proposal for IEEE 802.15.3a September 2004. http://www.wimedia.org/resources/resourcelibrary.asp?id=res.

[56] J. Y. Lee and R. Scholtz. Ranging in a dense multipath environment using an uwb radio link. *IEEE J. Select areas in communications*, 20(9):1677.

[57] C. Kuratli and Q. Huang. A cmos ultrasound range-finder microsystem. *IEEE Journal of Solid-State Circuits*, 35, December 2000.

[58] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *SenSys. ACM Press*, pages 50–61, 2004.

[59] D. W. Hanson. Fundamentals of two-way time transfer by satellite. In *43rd Annual Frequency Control Symposium*, pages 174–178, 1989.

[60] B. Razavi. *Design of Integrated Circuits for Optical Communications*. McGraw and Hill, New York, 2003.

[61] T. Pollet, M. van Bladel, and M. Moenclaey. BER sensitivity of ofdm systems to carrier frequency offset and wiener phase noise. *IEEE Trans. on Communications*, 43(2/3/4):191–193, Feb-Apr 1995.

[62] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 1999.

[63] J. Rabaey, A. Chandrakasan, and B Nikolic. *Digital Integrated Circuits*. Prentice Hall, 2003.

[64] M. J. Ammer, M. Sheets, T. Karalar, M. Kuulusa, and J. Rabaey. A Low-Energy Chip-Set for Wireless Intercom. In *Proceedings of the Design Automation Conference (DAC)*, Anaheim, CA, June 2-6 2003.

[65] Multi Carrier Multi Antenna research group. http://bwrc.eecs.berkeley.edu/research/MCMA/.

[66] T. Schmidl and D. Cox. Robust Frequency and Timing Synchronization for OFDM. *IEEE Transactions on Communications*, 45(12):1613–1621, December 1997.

[67] N. Golmie. Interference in the 2.4 ghz band. In *Proceedings of The First International Conference on Applications and Services in Wireless Networks, ASW 2001*, pages 187–199, Evry, France, July 25-27 2001.

[68] I. O'Donnell. *A Baseband, Impulse Ultra-Wideband Transceiver Front-end for Low Power Applications.* PhD thesis, UC Berkeley, May 2006.

[69] Mike S. W. Chen. *High-Speed, Low-Power A/D Converter for a Low-Cost UWB Sub-sampling Radio.* PhD thesis, UC Berkeley, May 2006.

[70] T. Terada, S. Yoshizumi, M. Muqsith, Y. Sanada, and T. Kuroda. A cmos ultra-wideband impulse radio transceiver for 1-mb/s data communications andpm2.5-cm range finding. *IEEE Journal of Solid-State Circuits*, 41:891–898, April 2006.

[71] J. Pehkonen and J. Kostamovaara. An integrated laser radar receiver with resonance-based timing discrimination. In *Proceeding of the 30th European Solid-State Circuits Conference, 2004. ESSCIRC 2004.*, pages 427–430, September 2004.

[72] N. Filiol et. al. A 22 mw bluetooth rf transceiver with direct rf modulation and on-chip if filtering. In *IEEE International Solid-State Circuits Conference, Digest of Technical Papers*, February 2001.

[73] C. Cojocaru, T. Pamir, F. Balteanu, A. Namdar, D. Payer, I. Gheorghe, T. Lipan, K. Sheikh, J. Pingot, H. Paananen, M. Littow, M. Cloutier, and E. MacRobbie. A 43mw bluetooth transceiver with -91dbm sensitivity. In *IEEE International Solid-State Circuits Conference, Digest of Technical Papers.*, February 2003.

[74] A. Zolfeghari, A. Chan, and B. Razavi. A 2.4ghz 34mw cmos transceiver for frequency hopping and direct-sequence applications. In *IEEE International Solid-State Circuits Conference, Digest of Technical Papers*, February 2001.

[75] Y. Chiu, B. Nikolic, and P. Gray. Scaling of analog-to-digital converters into ultra-deep-submicron cmos. In *Proceedings of the IEEE 2005 Custom Integrated Circuits Conference*, pages 375 – 382, September 2005.

Appendix A

CRLB calculations

Derivation of CRLB vs bandwidth $f = \frac{\omega}{2\pi}$ expression for low Signal to Noise Ratio (SNR) value of 3dB

$$Var[ToF] = \frac{1}{SNR}\left(1 + \frac{1}{SNR}\right)\frac{1}{\omega^2} \tag{A.1}$$

$$Var[ToF] = \frac{1}{2}\left(1 + \frac{1}{2}\right)\frac{1}{\omega^2} = \frac{3}{4 \times [2\pi f]^2} \tag{A.2}$$

$$\sigma_{ToF} = \sqrt{Var[ToF]} = \frac{\sqrt{3}}{4\pi f} \approx \frac{0.14}{f} \tag{A.3}$$

$$\sigma_{range} = \sigma_{ToF}\ c \approx \frac{0.14}{f \times 10^6}[300 \times 10^6] = \frac{42}{f}[m/MHz] \tag{A.4}$$

to achieve $\sigma_{range} = 1m$

$$f = \frac{42}{\sigma_{range}}[MHz] = 42MHz \tag{A.5}$$

Appendix B

Controller VHDL Description

```
ranger_fsm :  process(start,full,ext_count, done_fft,rep,rx_state,
    empty,del_ext_count, tx_setup,skip,fast_slo,
    fft,din_src,loop_over,div_done)
    begin -- process
        next_ext_count <= "0000000";
        start_fft <= '0';
        next_fft <= fft;
        next_rep <= rep;
        push_on_fifo <= '0';
        next_we <= '0';
        re <= '0';
        avg_we_n <= '1';
        next_new_data <= '0';
        next_div_count <= 0;
        next_tx_setup <= tx_setup;
```

```
            next_din_src <= din_src;

            next_fast_slo <= fast_slo; --for fifo modes

            next_skip <= skip;

            dividing <= '0';

            div_begin <= '0';

            fft2avg_mem <= '0';

            del_addr4fft <= '0';

            state_var <= "0000";

            searching <= '0';

            reset_max_n <= '1';


        case rx_state is

            when reset_st =>

                state_var <= "0001";

                if tx_setup = '1' then

                    next_rx_state <= begin_ifft;

                    next_fft <= '0';

                    next_fast_slo <= '0';

                elsif start = '1' then

                    next_rx_state <= digest_fs0;

                    next_fast_slo <= '1';

                else

                    next_rx_state <= reset_st;

                end if;

                next_din_src <= '0';

            when digest_fs0 =>
```

```vhdl
                state_var <= "0010";

                next_rx_state <= write_buffer;

                next_rep <= 0;

                next_we <= '1';
        when write_buffer =>

            if full = '0' then

                next_we <= '1';

                next_rx_state <= write_buffer;

            else

                next_rx_state <= digest_fs1;

                next_fft <= '1';

                next_fast_slo <= '0';

            end if;
        when digest_fs1 =>

            next_rx_state <= avg_fifo;


        when read_batch =>

            if empty = '1' then

                next_rx_state <= threshold;

            else

                next_rx_state <= read_batch;

                next_ext_count <= ext_count + 1;

                re <= '1';

            end if;
        when avg_fifo =>

            next_rx_state <= avg_fifo_wait;
```

```
            next_ext_count <= ext_count;

            re <= '1';

        when avg_fifo_wait =>

            if empty = '1' then

                next_rx_state <= begin_fft;

            else

                avg_we_n <= '0';

                next_rx_state <= avg_fifo;

                next_ext_count <= ext_count + '1';

            end if;

        when begin_fft =>

            next_rx_state <= read_fft_batch;

            start_fft <= '1';


        when read_fft_batch =>

            del_addr4fft <= '1';

            if del_ext_count = "1111111" then

                next_rx_state <= wait_fft;

            else

                next_rx_state <= read_fft_batch;

                next_ext_count <= ext_count + '1';

            end if;


        when wait_fft =>

            if done_fft = '1' then

                next_rx_state <= read_fft_result;
```

```
                    push_on_fifo <= '1';

                else

                    next_rx_state <= wait_fft;

                end if;


        when read_fft_result =>

            fft2avg_mem <= '1';

            if del_ext_count /= "1111111" then

                next_ext_count <= ext_count + '1';

                next_rx_state <= read_fft_result;

                avg_we_n <= '0';

                push_on_fifo <= '1';

            else

                next_rx_state <= divide;

                div_begin <= '1';

            end if;
-- the fft is finished here.....


-- now divide the signal with it's pilot spectrum.
        when divide =>

            if ext_count = "1111111" and div_done = '1' then

                next_rx_state <= begin_ifft_from_avg;

                next_fft <= '0';

                avg_we_n <= '0';

            elsif div_done = '1' then

                next_rx_state <= divide;
```

```
                next_ext_count <= ext_count + 1;

            div_begin <= '1';

            avg_we_n <= '0';

            else

                next_rx_state <= divide;

                next_ext_count <= ext_count;

            end if;

            dividing <= '1';


--> --read from the avg_mem for taking the ifft..

        when begin_ifft_from_avg =>

            next_rx_state <= read_ifft_from_avg;

            start_fft <= '1';

            del_addr4fft <= '1';


        when read_ifft_from_avg =>

            del_addr4fft <= '1';

            if del_ext_count = "1111111" then

                next_rx_state <= wait_ifft;

            else

                next_rx_state <= read_ifft_from_avg;

                next_ext_count <= ext_count + 1;

            end if;
-- start the ifft after reading from avg_memory..  -- only the

external written data is stored on the FIFO -- <--
```

```vhdl
-- -->in this part read the iff data from the pilot sequence.
        when begin_ifft =>

            next_rx_state <= read_ifft_batch;

            start_fft <= '1';

            re <= '1';

        when read_ifft_batch =>

            if ext_count = "1111111" then

                next_rx_state <= wait_ifft;

            else

                next_rx_state <= read_ifft_batch;

                next_ext_count <= ext_count + 1;

                re <= '1';

            end if;
-- <-- -- --> run the ifft..
        when wait_ifft =>

            if done_fft = '1' then

                next_rx_state <= write_cir;

                push_on_fifo <= '1';

--tentative

                next_din_src <= '1';-- din is from the fft

                output

            else

                next_rx_state <= wait_ifft;

            end if;
-- <-- end of run fft
```

```vhdl
    when write_cir =>

        if del_ext_count /= "1111111" then

            next_ext_count <= ext_count + 1;

            fft2avg_mem <= not tx_setup;

            push_on_fifo <= '1';

            next_rx_state <= write_cir;

        else

            next_rx_state <= waste_after_ifft;

            push_on_fifo <= tx_setup;

        end if;


    when waste_after_ifft =>

        next_tx_setup <= '0';

        next_rx_state <= threshold;

        next_din_src <= '0';

        reset_max_n <= '0';


    when threshold=>

        searching <= '1';

        if del_ext_count = "1111111" then

            next_rx_state <= finish_threshold;

        else

            next_rx_state <= threshold;

            next_ext_count <= ext_count + 1;

        end if;
```

```
        when finish_threshold =>

            next_rx_state <= idle;

            searching <= '1';


        when idle =>

            if loop_over = '1' then

                next_rx_state <= reset_st;

            else

                next_rx_state <= idle;

            end if;


    end case;

end process;
```